Modeling Musical Style with Language Models for Composer Recognition

María Hontanilla, Carlos Pérez-Sancho, and Jose M. Iñesta

University of Alicante, Spain
inesta@dlsi.ua.es
http://grfia.dlsi.ua.es

Abstract. In this paper we present an application of language modeling using n-grams to model the style of different composers. For this, we repeated the experiments performed in previous works by other authors using a corpus of 5 composers from the Baroque and Classical periods. In these experiments we found some signs that the results could be influenced by external factors other than the composers' styles, such as the heterogeneity in the musical forms selected for the corpus. In order to assess the validity of the modeling techniques to capture the own personal style of the composers, a new experiment was performed with a corpus of fugues from Bach and Shostakovich. All these experiments show that language modeling is a suitable tool for modeling musical style, even when the styles of the different datasets are affected by several factors.

Keywords: Music information retrieval, language models, musical style.

1 Introduction

Musical style is a quality of music that most people can perceive intuitively. It is often used to describe, categorize, and even compare songs and albums. Some authors have long been interested in it also as a tool for guiding the algorithmic composition process [1]. Although there is not a formal definition of what a musical style is, some authors have given some interesting definitions of the term:

- *Style is a replication of patterning, whether in human behavior or in the artifacts produced by human behavior, that results from a series of choices made within some set of constraints* [6].
- (Style is) *a recurring arrangement of features in musical events which is typical of an individual (composer, performer), a group of musicians, a genre, a place, a period of time* [4].

These two definitions coincide in that what makes a style is the repetition of some elements, and thus pattern recognition techniques seem to be the perfect tools to discover these patterns in order to model musical style. Moreover, the definition given by Fabbri fits perfectly the way this term has been used in the

J.M. Sanches, L. Micó, and J.S. Cardoso (Eds.): IbPRIA 2013, LNCS 7887, pp. 740–748, 2013.

Music Information Retrieval (MIR) literature, since many works in this area have focused in different aspects of musical style, including genre, composer, geographical origin, or historical periods, among others. A thorough review on the different uses of musical style in MIR tasks can be found in [9].

In a previous work [8] we used language modeling techniques to model the styles of different musical works in order to recognize their composers, with successful results. In this work we will focus on modeling the style of different composers. For this, we replicated the experiments conducted by van Kranenburg and Backer in [5], with a corpus of 5 composers from the Baroque and Classical periods.

1.1 Previous Works on Modeling Composer Style

Little work has been done in the modeling of composer styles, and it has been mainly done in the audio domain. Many of them can be found in the several editions of the Music Information Retrieval Evaluation eXchange (MIREX), where a *classical composer identification task* was proposed in 2007 and 2008. In this task, experiments were performed using a data set of 30-second audio clips from 11 composers: Bach, Beethoven, Brahms, Chopin, Dvořák, Händel, Haydn, Mendelssohnn, Mozart, Schubert, and Vivaldi.

In the symbolic domain, we can find the works by Ogihara and Li [7] and van Kranenburg and Backer [5], using harmonic and melodic information respectively.

In [7], the authors explore the capabilities of n-grams of chord progressions to characterize the style of several jazz musicians and The Beatles. Songs are encoded using n-gram profiles, where each n-gram is weighted using its relative duration measured in beats over the whole sequence. Then, the cosine of the product of two profiles is used as a similarity measure to study the separability between the different composers and their links, using a hierarchical clustering.

The authors also study different levels of chord information encoding, using chord triads, 6th and 7th chords, and extensions (9th, 11th and 13th). They conclude by selecting 20 *style markers* (4-grams of 7th chords) as the best to characterize the eight styles studied. However, no classification is performed to empirically support their conclusions.

Special attention will be paid to the work of van Kranenburg and Backer [5]. In this work, a set of features was developed based on musicological criteria. 20 features were selected by the authors that refer mainly to the polyphonic relationships between voices, as for example the vertical intervals weighted by duration, parallel motion, or dissonance treatment among others.

These features are extracted using a 30 bars sliding window. Then, a feature selection is performed in order to select the set of features that contribute the most to discriminate between the training data sets, made up of compositions from the catalogue of the candidate composers. Finally, all the windows extracted from the piece under study are classified using a nearest neighbor classifier, and the individual decisions are combined to reach a final decision.

This framework was tested with a data set of five composers: Bach, Händel, Telemann, Haydn, and Mozart [5], reaching classification rates between 79.4% and 95.2% using several configurations of classes. High error rates were due to the presence of Haydn and Mozart, which are composers with very similar styles.

One of the main drawbacks of this method is that the feature selection procedure must be repeated for each configuration of data sets, since not all the features perform the same to distinguish between different composers. And most important, those features are only useful if working with polyphonic compositions: bars that are not strictly polyphonic must be discarded during the encoding process.

In this paper, a different approach is proposed to overcome those drawbacks, using a general-purpose encoding method for melodies, while music analysis is performed using language modeling, a technique that has proven to be very effective in music classification [2,8].

2 Methodology

In this section the encoding method for the musical pieces is described, along with the language modeling technique used in order to analyze the resulting sequences.

2.1 Melodic Encoding

In order to avoid the need for a specific encoding for polyphonic music, a simple encoding for melodies was selected based on that proposed by Doraisamy and Rüger [3]. When using this encoding, pitch intervals and inter-onset duration ratios (IOR) of consecutive notes are computed using Equations 1 and 2 respectively.

$$I_i = Pitch_{i+1} - Pitch_i \qquad (i = 1, \ldots, n-1) \tag{1}$$

$$R_i = \frac{Onset_{i+2} - Onset_{i+1}}{Onset_{i+1} - Onset_i} \qquad (i = 1, \ldots, n-2) \tag{2}$$

Then, these values are mapped into alphanumeric characters (ASCII). This way, melodies are transformed into textual sequences, so they become a suitable input for the language modeling method explained in the next section. Two variants of this encoding have been used. The first one encodes the symbols for the pitch intervals and IORs of two consecutive notes together as a single word (coupled), while in the second variant (decoupled) these symbols are splitted in two different words.

2.2 Language Modeling Using n-grams

A language model is a probability distribution that assigns a probability to a progression of words $P(w_1, \ldots, w_k)$, so that the probability of each word in the sequence is dependent on its *context* $P(w_i | w_1, \ldots, w_{i-1})$.

Estimating the probabilities of such a model can be an arduous task, and maybe computationally unaffordable, when dealing with long sequences. This is why language models are often approximated using n-gram models. An n-gram is a sequence of n words in which the first $n - 1$ words are considered as the context. Thus, the estimated probability of a word w_i given a context is computed as $P(w_i|w_{i-n+1}, \ldots, w_{i-1})$.

In order to perform authorship attribution with the n-grams, a different language model must be constructed for each composer in the dataset. Each sequence (song) in the dataset is decomposed in n-grams of a fixed length n. Then, the probability of each different n-gram is computed as the probability of the last word given its context. This probability can be easily calculated by dividing the number of occurrences of the n-gram by the number of occurrences of its context in the given dataset:

$$P(w_i|w_{i-n+1}, \ldots, w_{i-1}) = \frac{\mathcal{N}(w_{i-n+1}, \ldots, w_i)}{\mathcal{N}(w_{i-n+1}, \ldots, w_{i-1})} \quad . \tag{3}$$

Once a language model is constructed for each composer, the probability that a new music piece $w = w_1, \ldots, w_k$ has been generated by model c is:

$$P_c(w) = \prod_{i=1}^{k} P_c(w_i|w_{i-n+1}, \ldots, w_{i-1}) \quad . \tag{4}$$

Thus, a test sample can be assigned to the most probable composer by following the risk minimization criterion, i.e. given the set of classes $\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\}$, each test sample is assigned to the class $\hat{c}$ of the model which holds $\hat{c} = \arg\max_c P_c(w)$.

Parameter Smoothing. Even when the training set is big enough to build a good language model, there can be situations where we can find words in a test sample that have not been seen previously. When such situation occurs, the probability of the n-grams containing those words is zero, thus causing the probability of the whole sequence being zero by the application of Equation (4).

To avoid this problem, it is common to use a procedure known as *smoothing*, in which a small probability is substracted from the set of known words, and then shared out among all unseen words. There are several techniques to calculate the optimal amount of probability that must be taken off, and what percentage of it must receive every unseen word.

A similar problem happens when a new sequence of words is found. This can be solved using a process known as backing-off in which the probability of a previously unseen sequence of words can be estimated using a lower order model built using $(n - 1)$-grams. In this work, linear interpolation was used for estimating the weights of each model of order n.

3 Experiments with 5 Composers

In order to evaluate the ability of the language models to capture the style of different composers, we tried to replicate the experiments previously performed by van Kranenburg and Backer [5], using the same corpus of musical works.

3.1 Dataset

The corpus used in this experiment was made up of works from five different composers: Bach, Telemann, Händel, Haydn, and Mozart. It was built by van Kranenburg and Backer [5] for a composer style classification task. All the files are polyphonic, containing several voices splitted in separate tracks. The total length of this corpus is over 23500 bars, with an average length of 86 bars per work. They can be grouped as follows:

- J. S. Bach: 28 cantata movements.
- J. S. Bach: 30 fugues from "The Well-Tempered Clavier".
- J. S. Bach: 11 movements from "The Art of Fugue".
- J. S. Bach: 6 movements from the violin concerts.
- G. F. Händel: 37 movements from the Concerti Grossi, op. 6.
- G. F. Händel: 14 movements from trio sonatas, op. 2 and op. 5.
- G. Ph. Telemann: 30 movements from the "Fortsetzung des Harmonischen Gottestdienstes".
- G. Ph. Telemann: 23 movements from the "Musique de table".
- F. J. Haydn: 49 movements from the string quartets.
- W. A. Mozart: 46 movements from the string quartets.

3.2 Results and Discussion

Using the corpus described above, we tried to model the styles of the five different composers using one language model built for each of them. Since this corpus is made up of polyphonic MIDI files with one track per instrument, in order to be able to apply the melodic language modeling technique explained in the previous section, all the tracks in the MIDI files were encoded separately and the resulting strings were concatenated as one single file. Due to the monophonic constrain in the strings we are processing, we need to prevent the occurrence of two or more notes playing together at the same time. For that, we have applied the skyline polyphony reduction algorithm [10] that has reportedly obtained good results in this task. It is based on the simple rule of keeping the note with the highest pitch when two or more are playing.

Two experiments were performed with this corpus. In the first one, pairwise classification was performed between all the classes, in order to find the best combination of encoding (coupled or decoupled) and n-gram length for this task. Leaving-one-out success rates are shown in Table 1. It can be seen that the best results were obtained using the decoupled encoding and 4-grams, with very high success rates for most of the pairs.

Next, another experiment was performed in order to compare this method with the one used in [5], with the same configuration of classes used in that work. The results for both methods are shown comparatively in Table 2. Although the results obtained with the n-grams were poorer for most of the datasets, they were quite good considering that a general purpose encoding has been used, compared to the other specialized polyphonic feature set, as it was discussed in Section 1.1.

Table 1. Success rates in pairwise classification

Dataset	Decoupled encoding			Coupled encoding		
	$n = 2$	$n = 3$	$n = 4$	$n = 2$	$n = 3$	$n = 4$
Bach vs. Händel	83.3	88.1	**88.9**	86.5	86.5	87.3
Bach vs. Haydn	85.5	93.6	**96.8**	91.9	93.6	93.6
Bach vs. Mozart	90.1	95.0	**97.5**	95.9	95.9	95.9
Bach vs. Telemann	88.3	94.5	**95.3**	94.5	93.8	93.8
Händel vs. Haydn	89.0	92.0	**94.0**	**94.0**	92.0	92.0
Händel vs. Mozart	85.6	**94.9**	92.8	93.8	93.8	92.8
Händel vs. Telemann	87.5	90.4	**93.3**	83.7	83.7	89.4
Haydn vs. Mozart	67.4	70.5	66.3	65.3	**74.7**	68.4
Haydn vs. Telemann	90.2	**98.0**	96.0	94.1	95.1	94.1
Mozart vs. Telemann	86.9	93.9	**98.0**	96.0	96.0	97.0

Table 2. Success rates using 4-grams and the decoupled encoding (left) compared with those obtained by van Kranenburg (right)

Dataset	4-grams	van Kranenburg
{Bach}, {Telemann}, {Händel}, {Haydn}, {Mozart}	78.8	**80.1**
{Bach}, {Telemann}, {Händel}	87.2	**93.0**
{Bach}, {Telemann, Händel}	88.3	**95.2**
{Bach}, {Telemann, Händel, Haydn, Mozart}	89.4	**94.0**
{Telemann}, {Händel}	**93.3**	91.6
{Haydn}, {Mozart}	66.3	**79.4**
{Telemann, Händel}, {Haydn, Mozart}	**95.0**	93.5

When looking at the figures in Tables 1 and 2, it seems reasonable to say that the language models built from the datasets have been able to capture the characteristic traits from the musical language of each composer or set of composers used in the experiments. However, there are some aspects regarding the composition of this corpus that should be taken into consideration when interpreting these results.

A closer look at the corpus shows that the works included for each composer have different musical forms than for the others, and the instrumentation is therefore also varied. The only exception to this are the works included for Mozart and Haydn, with both datasets made up entirely of string quartets, and it was precisely for this couple of composers where the lowest success rates were obtained. Although it is well known that these two classical composers had very similar styles, which makes it difficult even for expert listeners to distinguish between them, it is not possible to tell to which extent the works selected in this corpus are also influencing these low results. This fact arises some doubts on whether we are modeling composer styles or, on the contrary, what the models are really capturing are the differences in the writing for each musical form. Thus, an additional experiment was devised in order to test this hypothesis.

4 Bach vs. Shostakovich

Taking into account the conclusions drawn from the previous experiments, another dataset was built reducing the variability in order to verify if the language models are really able to distinguish between two different composers when using the same musical form: the fugue. The composers in this corpus are Johann Sebastian Bach (1685–1750), and the russian composer Dmitri Shostakovich (1906–1975). Although these two composers belong to distant periods in the history of music, the works selected from the Shostakovich catalogue are fugues which the author wrote in homage to Bach's work, so we assume that the different temporal context of both composers should not be a big influence in the results. On the other hand, we assume that this distance should make this task easier than the comparison of Haydn and Mozart.

4.1 Dataset

Between 1950 and 1951, Shostakovich wrote 24 preludes and fugues for the piano, in homage to the two series of 24 preludes and fugues written by Bach in "Das Wohltemperierte Clavier". All these 24 works from Shostakovich were included in the corpus. For the Bach dataset, 35 fugues were selected from "Das Wohltemperierte Clavier" and "Die Kunst der Fuge". More pieces were included for Bach than for Shostakovich, in order to compensate for the longer duration of the latter's works, so the language models were built using a similar amount of data. All the pieces in the corpus are contained in MIDI files, which have been manually curated in order to solve some musical inconsistencies detected.

4.2 Results and Discussion

The experiments in this section were performed using the decoupled encoding for melodies, and n-gram models of order 2, 3, and 4. In this case, best results were obtained for n-gram size 3, reaching a success rate of 96.6%. Table 3 shows the confusion matrix for this experiment.

Table 3. Confussion matrix for the experiment Bach vs. Shostakovich using the decoupled encoding and 3-grams

	Bach	Shostakovich	% success
Bach	35	0	100.0
Shostakovich	2	22	91.7
		Total % success	96.6

As it can be seen in the table, all Bach works were correctly classified, and only 2 works of Shostakovich were misclassified as Bach's. These excellent results show that the language models have been able to distinguish between these two composers, even when using the same musical forms. As it was said before, it is possible that this distinction is also due to the different temporal context of both composers. However, the works in the corpus were selected trying to minimize the effect of this, so it is our belief that only the differences in the author's own composing styles have been the determining factor to distinguish between them.

5 Conclusions

In this work we have tried to capture the style of different musical composers using language modeling techniques. In order to show the ability of n-gram models in this task, we replicated the experiments in [5] with the same corpus of 5 composers. From the results in this experiments, and looking at the composition of the corpus, it has come to light that it is very difficult to completely isolate the personal style of a composer. In our experiments we have used digital scores for the selected musical works. In these scores, the composers used a musical language marked by their own style, which is also influenced by a number of external factors: the temporal context of the composer, the type of work or musical form, or the purpose of the work to name a few.

However, in a new experiment using a corpus with fugues of Bach and Shostakovich, we have been able to (almost) isolate the characteristic traits of each composer using language models. In this experiment only one musical form was used, and we tried to minimize the factors affecting the composer's styles, so only their own personal traits marked the differences in the datasets. The excellent results obtained, with a 96.6% success rate, show that the language models have been able to capture each composer style.

Anyway, in both experiments this modeling technique has shown to be flexible enough to capture the characteristics for each dataset, even when grouping together heterogeneous works. Thus, we can conclude that language models are a suitable tool for modeling musical style, no matter what the definition of style is (genre, period, composer, . . .).

Acknowledments. This work was supported by the projects DRIMS (TIN2009-14247-C02) and the PROMETEO/2012/017. The authors want to thank the reviewers for their interesting and valuable comments on the paper.

References

1. Cope, D.: Computers and Musical Style. A-R Editions, Madison (1991)
2. Cruz-Alcázar, P.P., Vidal, E.: Two grammatical inference applications in music processing. Applied Artificial Intelligence 22(1&2), 53–76 (2008)
3. Doraisamy, S., Rüger, S.: Robust polyphonic music retrieval with n-grams. Journal of Intelligent Information Systems 21(1), 53–70 (2003)
4. Fabbri, F.: Browsing music spaces: categories and the musical mind. In: Proceedings of the IASPM Conference (1999)
5. van Kranenburg, P., Backer, E.: Musical style recognition - a quantitative approach. In: Parncutt, R., Kessler, A., Zimmer, F. (eds.) Proceedings of the Conference on Interdisciplinary Musicology (CIM 2004), Graz, Austria (April 2004)

6. Meyer, L.B.: Style and Music: Theory, History, and Ideology. University of Chicago Press, Chicago (1989)
7. Ogihara, M., Li, T.: N-gram chord profiles for composer style representation. In: Proceedings of the 9th International Conference on Music Information Retrieval, ISMIR 2008, Philadelphia, Pennsylvania, USA, pp. 671–676 (September 2008)
8. Pérez-Sancho, C., Rizo, D., Iñesta, J.M.: Genre classification using chords and stochastic language models. Connection Science 21(2), 145–159 (2009)
9. Pérez-Sancho, C.: Stochastic Language Models for Music Information Retrieval. Ph.D. thesis, Universidad de Alicante, Alicante, Spain (July 2009)
10. Uitdenbogerd, A., Zobel, J.: Manipulation of music for melody matching. In: Proc. of ACM Multimedia, Bristol, UK, pp. 235–240 (1998)