A Methodological Contribution to Music Sequences Analysis

Daniele P. Radicioni and Marco Botta
Università di Torino, Dipartimento di Informatica
C.so Svizzera 185, 10149, Turin, Italy
{radicion,botta}@di.unito.it

Abstract

In this paper we present a stepwise method for the analysis of musical sequences. The starting point is either a MIDI file or the score of a piece of music. The result is a set of likely themes and motifs. The method relies on a pitch intervals representation of music and an event discovery system that extracts significant and repeated patterns from sequences. We report and discuss the results of a preliminary experimentation, and outline future enhancements.

1 Introduction

In the last few years many efforts have been spent on music issues within the AI community. Two main tasks have been addressed, requiring "intelligent" and sophisticated strategies: music analysis and music performance. The first line of research investigated topics such as performer recognition [1], harmonic analysis [2], segmentation [3], whereas the second one aims at reproducing expressive music performance by means of artificial systems [4, 5].

Music analysis is a relevant task, in that it deeply affects our comprehension of music, as regards of composition, performance and listening. Music analysis is a challenging task: consider, e.g., that a significant part of professional music instruction is concerned with assisting the learner in "understanding" music for the different purposes of composition and performance.

In this paper we point out the problem of discovering repeated patterns, as a major issue for building systems for music analysis. It is commonly acknowledged that in Western Tonal Music repetition plays a fundamental role, and individuating themes and motifs is a fundamental step towards discovering higher order blocks, and their dependency relationships as well [6]. For example, looking at a fugue, one would individuate its main constituents (subject and countersubject), and then recognize their disperse episodes, generated through *imitation* and *transposition* [7]. Discovering significant repetitions in music has many applications, such as providing tools for indexing large music corpora and for content-based retrieval from music databases [8]. Moreover, in the area of computer assisted music analysis, both didactic tools and softwares for music performers and analysts would benefit from the discovery of the motifs underlying whole pieces.

Three main complexity factors increase the difficulty of the task. Meaningful motifs are interleaved with irrelevant gaps. Moreover, we are interested in discovering modified occurrences of motifs, as well. Lastly, we are interested in discriminating motifs from insignificant repetitions, whilst the greater part of repetitions in music are not *perceptually significant* [9]. These issues make the discovery of repeated patterns a challenging problem, and an interesting test-bed for automatic systems.

In this work we propose a novel approach to individuate themes and motifs. We show that the musical patterns discovery can be tackled via a Hierarchical Hidden Markov Model (HHMM) approach: the present system takes as input MIDI files and returns scores, where the patterns found are highlighted, and the corresponding MIDI files of the discovered themes.

The paper is organized as follows. First, we define the problem being solved; then we point out some similarities with other fields where one has to handle episodes represented as strings of symbols. Next, we briefly review the literature, and then introduce our system. We describe the music encoding adopted, and illustrate the system's basic features. Finally we report about a preliminary experimentation, discussing the obtained results.

2 Patterns in Music

We address the problem of recognizing the most significant motifs, and the problem of recognizing their disperse episodes as well. At all times composers adopted various techniques for composing music from few ideas; but it is from the Renaissance that these techniques were refined to such an extent that it was possible to build entire pieces from a single musical idea. The simpler form of repetition is *literal* repetition, but far more often the repetition is combined with *variation*: changes may involve harmony and/or melody and/or rhythm. Among the most widely used variation techniques, we mention *augmentation* and *diminution* (where the length of the repeated notes is prolonged or shortened), *inversion* of intervals between note pairs and *contrary motion* [10] (Fig. 1). These techniques have been used by both historic and contemporary composers (see, e.g., the works from J.S. Bach (1685–1750) and A. Schoenberg (1874–1951)), in particular for polyphonic music. Polyphonic music can be defined as a texture consisting of overlapped lines, where each of the individual parts is called *voice*, even if it is played by instruments. Each voice is independent from the other ones (we don't have a melody together with the accompaniment, but rather equally important voices), thus retaining its identity (Fig. 2).

The problem of finding repeated patterns in polyphonic music can be formulated in the following terms. A motif can be thought of as a complex event (CE),

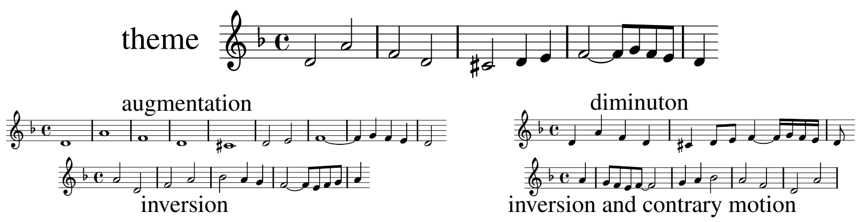

Figure 1: Main theme from *Contrapunctus* 4 from J.S. Bach's *Die Kunst der Fuge*, and the variations obtained through the techniques of inversion, inversion combined with contrary motion, augmentation and diminuton.

occurring sparsely in a sequence of notes. Such a complex event is composed by *episodes* of atomic events (AEs). In turn, episodes are composed by strings of symbols: in our case, by pitch intervals (see below, Section 4.1). In this setting, based on the properties of regular expressions, it is possible to infer a model of the motif being searched via an abstraction mechanism. This approach transports to the musical domain a well established modeling framework, which has been successfully tested on various domains, such as user (typing) profiling [11, 12].

3 State of the Art

Several algorithms that tackle the task of pattern recognition in music have been developed. We briefly review the principal and closely related ones, whilst a richer tour is provided in [9]. Dovey [13] proposes an algorithm for querying musical databases, where music events are represented as strings of (sets of) note pitches. Different kinds of relationships can be individuated between events, such as about harmony, or according to whether the two events under consideration fall into some pitch range. This approach also requires to specify the dimension of gaps between consecutive events. The pattern discovery technique devised by Conklin & Anagnostopoulou [14] relies on a representation based on a set of parameters (*viewpoints*, in the authors terms [15]): here each string represents an individual parameter, and meaningful repetitions are individuated based on a Hidden Markov Model, looking for repetitions more frequent than expected. These approaches only discover *exactly* repeated factors in strings.

On the other side, the algorithm by Rolland [16], which allows retaining much musical information such as duration, interval, degree in the overall tonality, has been designed to discover *approximately* repeated patterns. The author defines a similarity metrics between pairs of sequence segments: the Multi-Description Valued Edit Model implements a function for computing the *edit distance* between strings [17]. This algorithm allows discarding as not similar all the patterns below some threshold k, thus allowing to define a notion of

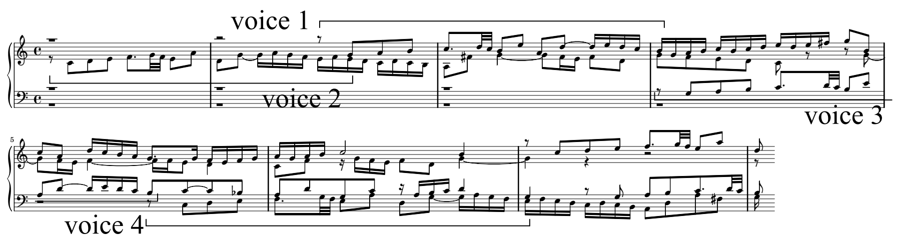

Figure 2: Bars 1–7 from Fugue in C major from Book 1 of J.S. Bach's *Das Wohltemperierte Klavier*: each voice states the subject.

k-similarity. Then, the distance between each pattern instantiation and the pattern itself is computed. Based on this proximity score, Rolland obtains a list of the most prominent repeated patterns. Cambouropoulos et al. [18] propose an algorithm for approximate string matching, by transporting to the music domain the well-known concept of *edit distance*.

Lartillot [19] devised an interesting mechanism for pattern induction, based on a plain encoding of pitch intervals and metrical salience. The algorithm first analyses note pairs within a temporal window of fixed size. At this step similar intervals are retained as potential patterns. Then, Lartillot's algorithm checks the prosecution of repeated patterns by considering only *melodic contour* (sequences being considered can prosecute upward, downward, or constant). This way, individuating approximate repetitions is no longer an issue. Also, this system is provided with a sort of abstraction mechanism, defining pattern *occurrences* as instances of pattern *classes*. However, a limitation of the approach resides in the fact that the algorithm cannot handle long sequences, due to the high computational cost necessary for this kind of analysis.

One popular approach for discovering repeated patterns is the"geometric" approach by Meredith et al. [9]: music is represented as a multidimensional dataset. Each event in the score can be encoded via an arbitrary number of dimensions, such as onset time, pitch, duration and voice. The authors define perceptually relevant repetitions in terms of the maximum pattern that can be translated into another pattern in the dataset. This is equivalent to finding out all the transposition-invariant occurrences of a pattern, according to a given dimension/set of dimensions. Unlike the above mentioned works, this approach allows handling polyphonic music.

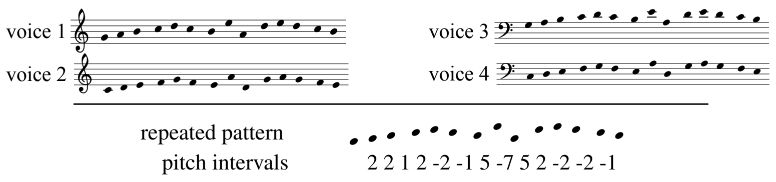

Figure 3: Top: the subject of the fugue presented in Fig. 2 is extracted from each voice. Bottom: all the voices exhibit the same shape, despite the notes are actually different. Commonalities among the different occurrences of the subject can be seen via a pitch interval representation.

4 The System

4.1 Encoding musical information

We take as input pieces encoded as standard MIDI files[1], where each musical "voice" is represented as a separate track: given n voices, we have as many tracks. For example, the piece in Fig. 2 can be encoded with 4 tracks (Fig. 3). We then extract from each track a sequence of music events: an event has a pitch, onset and offset; each new onset determines a new event.

Underpinned by music analysis [7] and music cognition [6] literatures, we are primarily concerned with music *intervallic* content, in that it can reveal pitch contour commonalities between motifs. For example, let us consider the excerpt presented in Fig. 2. Here we have a "curve" shaped motif (properly, the subject of the fugue), that accomplishes an ascending–jump–descending melodic movement. The essence of this kind of motifs –see e.g., the top of Fig. 3– is best grasped if one considers the pitch intervals between each pair of notes, as it is shown in the bottom of Fig. 3. We can see that the four occurrences of the subject are invariant under transposition, so that the pattern can be individuated considering the intervals rather than the actual notes. More sophisticated kinds of representation have been designed, such as Conklin's *multiple viewpoints* [15], that in principle could better fit to pattern discovery. However, as it was pointed out by Conklin [14], a systematic experimentation over 185 J.S. Bach chorales provided evidence that patterns were mainly found via "melodic intervals". As a consequence, we chose a simple representation accounting for intervals only; this has the advantage of permitting to translate music input into a string with a smaller alphabet.

4.2 Extracting and modeling motifs

In order to accomplish this step, we used a recently proposed event discovery system [11, 12]. The main idea is that of modeling each motif by means of a

[1] `http://www.midi.org/`.

Profile Hidden Markov model (PHMM), and representing a sequence of motifs interleaved with gaps by a Hierarchical Hidden Markov model (HHMM).

A Hidden Markov Model (HMM) is a stochastic finite state automaton [20] defined by a tuple $\lambda = \langle Q, O, A, B, \pi \rangle$, where:

- Q is a set of states, and O is a set of atomic events (observations),
- A is a probability distribution governing the transitions from one state to another. Specifically, any member $a_{i,j}$ of A defines the probability of the transition from state q_i to state q_j, given q_i.
- B is a probability distribution governing the emission of observable events depending on the state. Specifically, an item $b_{i,j}$ belonging to B defines the probability of producing event O_j when the automaton is in state q_i.
- π is a distribution on Q defining, for every $q_i \in Q$, the probability that q_i is the initial state of the automaton.

The difficulty, in this basic formulation, is that, when the set of states Q grows large, the number of parameters to estimate (A and B) rapidly becomes intractable. Nevertheless, the size of the parameters to estimate can be strongly reduced if one defines a structure for the automaton, where a number of state transitions and the possible emissions in a state are cut a priori. This corresponds to setting to 0 the corresponding values in matrix A and B. Actually, many work related to HMM applications consists in handcrafting the a priori structure of the automaton, as, for instance, the Profile Hidden Markov Model [21], widely used for DNA analysis.

A Profile Hidden Markov Model assumes that a motif has a canonical instantiation form, that can be affected by insertion and deletion errors. A PHMM is basically a forward graph with three categories of states:

- *match* states, where the emission corresponds to the unique symbol expected in the canonical instantiation;
- *insertion* states, where a number of symbols due to random noise can be inserted;
- *deletion* states, where non emission occurs where it was supposed to.

In addition, other two non emitting states are required: the *start* state, and the *end* state. Recursion is considered only in the form of self-loops associated to *insertion* states. It has been experimentally shown that PHMM is more accurate than string matching to detect motifs [21], and then we expect that it holds for musical sequences as well. Moreover, the algorithmic complexity inherent to PHMM is linear both for Viterbi and Forward-Backward algorithms.

The Hierarchical HMM (HHMM) proposed by Fine, Singer and Tishby [22] is an extension of the basic HMM, that immediately follows from the regular languages property of being closed under substitution; this property allows a large finite state automaton to be transformed into a hierarchy of simpler ones. More specifically, a HHMM is a hierarchy where, by numbering the hierarchy levels with ordinals increasing from the lowest towards the highest level[2], observations generated in a state ${q_i}^k$ by a stochastic automaton at level k are sequences generated by an automaton at level $k - 1$. The emissions at the lowest levels are again single tokens as in the basic HMM. Moreover, no direct transition may occur between the states of different automata in the hierarchy. As in HMM, in every automaton the transitions from state to state is governed by a distribution A and the probability for a state being the initial state is governed by a distribution π. The restriction is that there is only one state which can be the terminal state.

The major advantage provided by the hierarchical structure is a strong reduction in the number of parameters to estimate. In fact, automata at the same level in the hierarchy do not share interconnections: every interaction through them is governed by transitions at the higher levels.

A second advantage is that, as it will be described in the following, the modularization enforced by the hierarchical structure allows the different automata to be modified and trained individually, thus providing a natural subproblem decomposition.

The basic learning algorithm, fully described in [11, 12] builds the HHMM hierarchy bottom-up starting from the lowest level (actually, only two levels are built). The first step consists in searching for possible motifs, i.e., short chains of consecutive symbols that appear frequently in the learning traces, by means of classical methods used in DNA analysis [21]. A PHMM is then built from the found motifs. As models of the motifs are constructed independently from one another, it may happen that models for spurious motifs are constructed. At the same time, it may happen that relevant motifs are disregarded just because their frequency is not high enough. Both kinds of problems will be fixed at a later time. Starting from the models found at this step, a HHMM can be built in the following way: the input sequences are abstracted (i.e., rewritten in terms of the models found) by substituting each occorrence of a PHMM with a symbol in a new alphabet and subsequences between two motifs not attributed to any PHMM, by means of a special symbol called *gap*.

After this basic cycle has been completed, an analogous learning procedure is repetead on the abstracted sequences, where models are now built for sequences of *episodes*, searching for co-occorrent motifs. In this process, spurious motifs not showing significant regularities can be discarded. The major difference with respect to the first step, is that models built from the abstracted sequences are observable Markov models. This makes the learning task easier and decreases its computational complexity. After building the HHMM structure in this way, it can be refined by using standard training algorithms [23, 22].

[2]We use a reverse numeration for hierarchy levels, with respect to the original formulation in [22].

4.3 Producing the results

Once the HHMM has been built, we are ready to produce the final results of the system. In this last step, the acquired model is instatiated on every voice and two outputs are generated: a score of the piece tagged with the motifs found in that voice. In order to provide also audible results, we generate as many MIDI files as the found motifs (i.e., for every motif we produce a MIDI file that contains the notes in the motif). Both scores and MIDI files are available on the Web[3].

5 Experimental validation

In order to assess the results provided by the system, we have collected a dataset composed by the fugues from Book 1 of J.S. Bach's *Das Wohltemperierte Klavier*. We used MIDI recordings retrieved from BachCentral.com[4]. Due to problems encountered while processing the MIDI files[5], we could actually use only 15 of the 24 fugues from the original corpus.

We have run the system to discover the main motif (i.e., the subject) of each fugue. Also, we have been looking for all the occurrences of such subject throughout the fugue: specifically, we have been looking for the occurrences of the whole (if varied) subject, thus disregarding episodes such as the *stretti*, where only the head or the tail of the subject occurs. While testing the system on fugues motifs only, we considered motifs longer than a fixed amount (presently, this was set to 6 notes), thereby retaining motifs that have a score higher than the average of the scores left. The experimentation should answer the following questions: *i*) How much do such filtered data fit to the actual subjects; *ii*) How many of the subject repetitions do we find; *iii*) How many notes in the (repeated) subjects are actually individuated.

i) The system correctly recognized the subject in 72.55% of cases. Overall *ii*), 74.73% of the total number of subject occurrences were identified; also, on average, *iii*) we have individuated 80.86% of the notes in the subjects.

5.1 Discussion

The present results hardly compare with literature, because, to the best of our knowledge, no researchers addressing the pattern discovery in the musical domain have tested their systems in a systematic way. Rather, examples of individual patterns have been provided, or the most discriminative musical features have been pointed out –e.g., pitch intervals, contour– (see, respectively, the state of art systems presented in [9] and [14]). Reasonably, providing crude numbers would result improper in some cases, but perhaps the lack of system-

[3]`http://www.di.unito.it/~botta/bach-fugues`

[4]`http://www.bachcentral.com/wtcMidi1.html.`

[5]For reading the files and extracting the tracks, we used the parsing routines provided by the standard javax.sound.midi package.

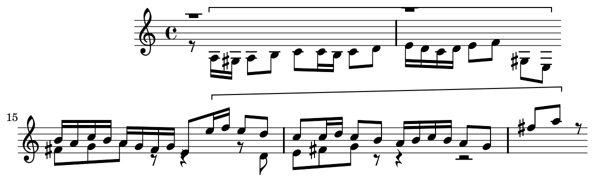

Figure 4: Top: the subject of Fugue 20; bottom: the subject is stated by the *soprano* voice with the inversion of intervals.

atic experiments witnesses about the difficulty of the task, as well. Hence our results can furnish a baseline against which other systems can be compared.

Provided that this preliminary experimentation considered only a small dataset, and that we have performed only a reduced form of automatic analysis, aiming at discovering vividly *individuated* motifs, the results are satisfying.

A closer look to the data may be helpful in completing the assessment of the results, and in suggesting some criteria for future improvements. In the test *i*) we identify the subject in 72.55% of cases, but if we retain all the results without filtering, then the success ratio raises to 100% of cases. Therefore, taken for granted that models for subjects can be acquired, refinements to the filtering function are at hand. For what concerns test *ii*), some improvements would be possible by considering additional music information, such as, e.g., the *absolute* intervals. Most likely this would be useful to discover occurrences of the subject with *inverted* (Fig. 1) intervals. Grasping these cases (Fig. 4) would further improve the system's accuracy. The test *iii*) would benefit from a slightly different splitting of the string in input.

However, the results also show that much work has still to be done. Many errors were committed when encountering "too short" subjects (e.g., Fugue 4), where it is harder to acquire a significant pattern. Moreover, we were penalized from cases where "real answer" significantly transforms the expositions of the subject (e.g., Fugue 21), thus increasing the difficulty to individuate the common underlying model.

6 Conclusions

This paper has presented a methodology for addressing a captivating problem: analyzing the *horizontal* textures of music. Namely, we have applied it for discovering the subjects in some fugues from J.S. Bach's *Das Wohltemperierte Klavier*. A working system implements the methodology: based on a plain though effective encoding of music, it acquires a model of the most significant repeated patterns. Then it outputs the results both as analyzed scores and as MIDI files, containing the patterns discovered. The experimental validation demonstrated the adequacy of the approach, while also pointing out some open issues, that will steer our future work.

Furthermore, the hierarchical hidden Markov model learned by the system could provide useful insights about the overall structure of musical pieces, allowing to describe them in terms of basic building blocks (motifs and gaps) and their relationships.

Acknowledgments

This work has been partly funded by the University of Piemonte Orientale that supported Daniele Radicioni. We would also thank Attilio Giordana and Ugo Galassi for working to the core system.

References

[1] Stamatatos, E., Widmer, G.: Automatic identification of music performers with learning ensembles. Artificial Intelligence **165** (2005) 37–56

[2] Conklin, D.: Representation and discovery of vertical patterns in music. In Anagnostopoulou, C., Ferrand, M., Smaill, A., eds.: Music and Artificial Intelligence: Procs. ICMAI. Volume 2445 of LNAI., Springer-Verlag (2002) 32–42

[3] Bod, R.: A Unified Model of Structural Organization in Language and Music. Journal of Artificial Intelligence Research **17** (2002) 289–308

[4] Widmer, G.: Modeling the Rational Basis of Musical Expression. Computer Music Journal **19**(2) (1995) 76–96

[5] Lopez de Mantaras, R., Arcos, J.: AI and Music: From Composition to Expressive Performance. AI Magazine **23** (2002) 43–57

[6] Lerdahl, F., Jackendoff, R.: A Generative Theory of Tonal Music. MIT Press, Cambridge, MA (1983)

[7] Bent, I., ed.: Music Analysis in the Nineteenth Century. Cambridge University Press, Cambridge (1994)

[8] Hsu, J.L., Liu, C.C., Chen, A.: Efficient Repeating Pattern Finding in Music Databases. In: Procs. of International Conference on Knowledge and Information Management. (1998) 281–288

[9] Meredith, D., Lemström, K., Wiggins, G.: Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. Journal of New Music Research **31**(4) (2002) 321–245

[10] de la Motte, D.: Kontrapunkt. Ein Lese und Arbeitsbuch. Kassel, Bärenreiter (1981)

[11] Botta, M., Galassi, U., Giordana, A.: Learning Complex and Sparse Events in Long Sequences. In: Procs. of the 16th European Conference on Artificial Intelligence, Valencia, Spain (2004) 425–429

[12] Galassi, U., Giordana, A., Saitta, L., Botta, M.: Learning Regular Expressions from Noisy Sequences. In: Procs. of 6th International Symposium on Abstraction, Reformulation and Approximation, Edinburgh, Scotland (UK) (2005) 92–107

[13] Dovey, M.: A Technique for "regular expression" style searching in polyphonic music. In: Procs. of the Internat. Music Information Retrieval Conference. (2001)

[14] Conklin, D., Anagnostopoulou, C.: Representation and Discovery of Multiple Viewpoint Patterns. In: Procs. of the 2001 International Computer Music Conference. (2001) 479–485

[15] Conklin, D., Witten, I.H.: Multiple viewpoint systems for music prediction. Journal of New Music Research **24**(1) (1995) 51–73

[16] Rolland, P.Y.: Discovering Patterns in Musical Sequences. Journal of New Music Research **28**(4) (1999) 334–350

[17] Navarro, G.: A Guided Tour to Approximate String Matching. ACM Computing Surveys **33**(1) (2001) 31–88

[18] Cambouropoulos, E., Crochemore, M., Iliopoulos, C., Mouchard, L., Pinzon, Y.: Algorithms for Computing Approximate Repetitions in Musical Sequences. Intern. J. Computer Math. **79**(11) (2002) 1135–1148

[19] Lartillot, O.: Discovering Musical Patterns through Perceptive Heuristics. In: Procs. of International Symposium on Music Information Retrieval. (2003)

[20] Rabiner, L.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of IEEE **77(2)** (1989) 257–286

[21] Durbin, R., Eddy, S., Krogh, A., Mitchison, G.: Biological sequence analysis. Cambridge University Press (1998)

[22] Fine, S., Singer, Y., Tishby, N.: The hierarchical hidden markov model: Analysis and applications. Machine Learning **32** (1998) 41–62

[23] Murphy, K., Paskin, M.: Linear time inference in hierarchical hmms. In: Advances in Neural Information Processing Systems (NIPS-01). Volume 14. (2001)