ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-1077

Imitative Leadsheet Generation with User Constraints

François Pachet and **Pierre Roy**[1]

Abstract. We introduce the problem of generating musical leadsheets, i.e. a melody with chord labels, in the style of an arbitrary composer, that satisfy arbitrary user constraints. The problem is justified by the very nature of musical creativity, as many composers create music precisely by imitating a given style to which they add their own constraints. We propose a solution of this problem by formulating it as a Markov constraint problem. Markov constraints enable users to create stylistically imitative leadsheets that satisfy a large palette of constraints. We show that generated leadsheets are stylistically consistent by reclassifying them using Markov classifiers.

1 THE PROBLEM

Leadsheets are music scores used in many genres of popular music such as jazz, Pop or songwriting. Leadsheets are composed of a monophonic melody, usually made up of simple rhythms, and a sequence of corresponding chord labels (Figure 1). Many works in Artificial Intelligence and Computer music have addressed the issue of generating musical material given some information about the structure of the expected result, such as figured bass, soprano given or harmonization of existing melodies [1, 4]. However, none, to our knowledge, have focused on the issue of generating leadsheets from scratch, that satisfy both style constraints (sound like a given author or corpus) as well as arbitrary user defined constraints.

Figure 1. An extract of the leadsheet for Very Early by Bill Evans.

Leadsheet generation raises specific issues. The difficulty is that the synchronization between the melody and chord labels is unknown a priori, because chord labels are attached to bars (in the example in Figure 1, there is one chords per bar), but bars may contain an arbitrary number of notes. The total duration of notes in a bar is fixed (e.g. 3 quarters). In Figure 1, bars contain 1, 2 or 3 notes. Informally, a leadsheet consists of two "parallel" sequences: one that contains chord labels and one that contains notes. A chord label is a triplet $\langle R, T, d\rangle$, where R is a pitch-class, e.g., C, and T is a chord type, e.g., 'major', '7', and d is the duration (expressed in, e.g., number of beats). A note is a pair $\langle p, d\rangle$, where p is the pitch, e.g., a midi-pitch between 0 and 127, and d is the duration of the note. A temporal sequence of chord labels (resp. notes) is a sequence of non-overlapping consecutive chord labels (resp. notes). A leadsheet L is a pair $\langle C, N\rangle$, where $C = (C_1, \dots, C_k)$ is a temporal sequence of chord labels and $N = (N_1, \dots, N_l)$ is a temporal sequence of notes. C and N have the same total duration. We represent temporal positions as ordered pairs $t = \langle b, p\rangle$, where b is the bar index and p is the beat position in the bar. For a temporal position t in a leadsheet L, the transition $\tau_L(t)$ is the 4-tuple $\langle C_t^-, C_t^+, N_t^-, N_t^+\rangle$, where C_t^- (resp. N_t^-) is the last chord label (resp. note) starting before or at t and C_t^+ (resp. N_t^+) is the first chord label (resp. note) ending after or at t.

Let $\mathcal{L} = \{L_1, \dots, L_n\}$ be a set of leadsheets (called a corpus), the **basic imitative leadsheet generation** problem consists in generating a new leadsheet $L = \langle C, N\rangle$ such that:
$\forall t = \langle b, p\rangle$, the transition $\tau_L(t)$ occurs in at least one L_i, i.e., $\exists i, \exists t' = \langle b', p'\rangle$, such that $\tau_{L_i}(t') = \tau_L(t)$ and such that $p' = p$ (the element is at the same metrical position as the original one).
User constraints may be added to generate leadsheets satisfying specific properties. For instance, let d be a duration, e.g., 32 bars. The **duration constrained leadsheet generation** problem over $LGP(\mathcal{L}, d)$ consists in generating a new leadsheet $L = \langle C, N\rangle$ such that the duration of L is d. Let d_{max} be a duration, **non-plagiaristic leadsheet generation** can in turn be formulated as: $L \notin \mathcal{L}$ and the longest fragment of L that appears verbatim in some L_i is shorter that d_{max}.

2 MARKOV CONSTRAINTS

Markov models have long been used to generate music in the style of a composer [2, 7]. Markov models can be estimated easily from arbitrary musical corpora, and random walks in those Markov models can be used to generate convincing musical material. However, these techniques are notoriously difficult to control, so generated artefacts cannot enforce a priori musical properties. The techniques of Markov constraints have been introduced to deal precisely with the issue of generating sequences from a Markov model estimated from a corpus, that also satisfy non Markovian, user defined properties. The generation of Markov sequences is formulated as a combinatorial sequence generation problem. User constraints are represented using the huge palette of constraints available, and Markovianity is represented as specific constraints [6, 8]. Recent developments have focused on constraints for which efficient filtering procedures can be designed, including unary constraints [11], meter [12] or maxOrder [9].

[1] Sony Computer Science Laboratory, Paris, France, email: pachetcsl@gmail.com

3 VIEWPOINTS

To estimate a Markov model, the first step is to produce sequences from training data. Indeed, musical elements composing leadsheets are complex objects: a note is made up of a pitch (integer value), duration (float), starting beat (float), and possibly much more. The projection of the note is usually called a viewpoint [3]. The decision concerning which viewpoint to use for a given musical task influences considerably the quality of the outcome, to an extent that is not yet fully understood. In our case, we model leadsheet elements by a viewpoint, called here a chunk, including both metrical information (position in bar) and harmonic information (underlying chord label). Figure 2 shows an extract of a leadsheet and its corresponding viewpoint representation as a sequence of strings.

Figure 2. An extract of the leadsheet in Figure 1 with the corresponding segmentation and viewpoint structure. The sequence of viewpoints is here *pitch*, *position in bar*, *duration*, *chord label*): {G2, 1, 2, CM7}, {E2, 3, 1, CM7}, {Bb1, 1, 1, Bb7}, etc.

4 LEADSHEET GENERATION

Once leadsheets are transformed into chunk sequences, Markov constraints can be used to control leadsheet generation. For instance, unary constraints [11] can be set to ensure that the first note starts on the first beat (no silence at the beginning), or/and that the last note ends on the last beat of a bar, or on specific chord labels. However, with only unary constraints, the total duration of the piece cannot be controlled, as only the number of chunks is specified. Harmony-wise, the user can control chords by imposing arbitrary properties on chords, specified by their index. No information is available concerning the metrical position of chords.

With the meter constraint [12] the user can control the harmonic structure metrically, i.e. impose arbitrary properties holding at specific metrical positions in the piece. For instance, one can post harmonic constraints at the beginning of a specific bar, regardless of the number of notes in the bar or elsewhere in the sequence.

Non-plagiarism can be handled efficiently with the maxOrder constraint [9]. Figure 3 shows a generated leadsheet in the style of Bill Evans that guarantees that no chunk of duration more than 6 beats is reused from the original corpus (all Bill Evans songs in ¾).

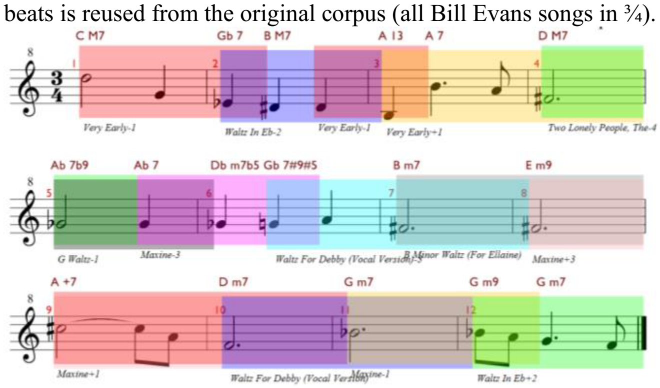

Figure 3 A generated leadsheet in the style of Bill Evans, in 3/3, with a constraint on the first (C 7) and last chords (G 7), and with a max order of 4 beats. Colors indicate chunk contiguity in the corpus. Labels indicate the origins of chunk in the corpus.

5 EVALUATION

We have used our system on a comprehensive leadsheet database [10]. A fundamental question is to which extent generated leadsheets can be considered indeed to be in the style of the original training corpus. Informal experiments with trained musicians show that users definitively recognize their favorite composers, but this kind of experiment is biased due to the unknown amount of knowledge users have of the training corpus. We have conducted experiments consisting in reclassifying generated leadsheets using a Markov classifier [5], trained on a selection of leadsheets from a predefined group of composers. The results, not given here for reason of space, show that leadsheets generated from a composer are classified as that composer with accuracy comparable to that of a train/test procedure on original songs. Interestingly, adding global constraints such as *meter* or *maxOrder* do not degrade the classification significantly. This shows that style imitation with Markov constraints is robust enough to be considered as a basis for computer-assisted music composition. Current work focuses on the development of constraints to impose specific genre-dependent structure (e.g. AABA, Blues, etc.). Examples of generated leadsheets in the style of many jazz composers as well as a web radio can be found at www.flow-machines.com/leadsheetGeneration.

ACKNOWLEDGEMENTS

This research is conducted within the Flow Machines project funded by the ERC under the European Union's 7th Framework Programme (FP/2007-2013) / ERC Grant Agreement n. 291156. We thank Jeff Suzda and Mark d'Inverno for their enthusiastic feedbacks on the generated leadsheets.

REFERENCES

[1] Anders, T., E. Miranda (2011). A Survey of Constraint Programming Systems for Modeling Music Theories and Composition. ACM Computing Surveys. 43(4).

[2] Brooks, F. P., Hopkins, A. L., Neumann, P. G., & Wright, W. V. (1993). An experiment in musical composition. In S. M. Schwanauer & D. A. Levitt (Eds.), Machine models of music, Cambridge: MIT.

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

[4] Ebcioglu, K. (1990) An Expert System for Harmonizing Chorales in the Style of J. S. Bach. J. Log. Program. 8(1): 145-185.

[5] Hedges, T. Roy, P. and Pachet, F. (2014) Predicting the Composer and Style of Jazz Chord Progressions, J. of New Music Research, special issue on Music and Machine-Learning.

[6] Morin, M. and Quimper, C.-G. The Markov Transition Constraint. Proc. of the 11th CP-AI-OR, 2014.

[7] Nierhaus, G. (2009), Algorithmic composition—paradigms of automated music generation, Berlin: Springer.

[8] Pachet, F. and Roy, P. Markov constraints: steerable generation of Markov sequences. Constraints, 16(2):148-172, March 2011.

[9] Papadopoulos, A. Roy, P. and Pachet, F. Avoiding Plagiarism in Markov Sequence Generation, AAAI, Quebec, Canada, 2014.

[10] Pachet, F., Suzda, J., Martin, D. (2013) A Comprehensive Online Database of Machine-Readable Lead Sheets for Jazz Standards. Proc. of ISMIR, pp. 275-280, Curitiba (Brazil).

[11] Pachet, F., Roy, P. and Barbieri, G. Finite-Length Markov Processes with Constraints. IJCAI, pp. 635-642, Barcelona, Spain, July 2011.

[12] Roy, P. and Pachet, F. Enforcing Meter in Finite-Length Markov Sequences. AAAI, 2013.