ATIAM 2019 - ML Project
Introduction of musical distances for multi-step inference of jazz chord progressions

Tristan Carsault[1], Jérôme Nika[2], Philippe Esling[1*]
[1]Institut de Recherche et Coordination Acoustique Musique (IRCAM)
UPMC - CNRS UMR 9912 - 1, Place Igor Stravinsky, F-75004 Paris
[2]Le Fresnoy - Studio national des arts contemporains
{carsault, jnika, esling}@ircam.fr

Abstract

This project aims to predict chord progressions of jazz music, represented as coherent chord label sequences with the help of probabilistic models. In this study, we propose to use different neural network models to generate symbolic chord sequences. Besides, we study the impact of the introduction of different musical distances through the loss function during the training of our models. Thus, we want to improve existing methods by doing multi-step prediction and by injecting music theory knowledge through the learning method in order to be able to perform accurate prediction of chord sequence and jazz melody generation. Ultimately, this project could be used to perform automatic accompaniment and improvisation.

1 Introduction

Most of today's Western music is based on an underlying harmonic structure. This structure describes the progression of the piece with a certain degree of abstraction and varies at the scale of the pulse. It can therefore be represented by a "chord sequence", with a chord representing the harmonic content of a beat. Hence, real-time music improvisation system, such as Nika et al. [2017], crucially need to be able to predict chords in real time along with a human musician at a long temporal horizon. Indeed, chord progressions aim for definite goals and have the function of establishing or contradicting a tonality. A long-term horizon is thus necessary since these structures carry more than the step-by-step conformity of the music to a local harmony. Specifically, the problem can be formulated as: given a history of beat-aligned chords, output a predicted sequence of future beat-aligned chords *at a long temporal horizon*. In this project, we use a set of ground truth chord sequences as input, but the models developed here could be combined with an automatic chord extractor for use in a complete improvisation system.

In this study we use the term multi-step chord sequence generation for the prediction of a series of possible continuing chords according to an input sequence (Carsault et al. [2019]). Furthermore, we propose to exploit specific properties and relationships between chord labels in order to improve the learning of statistical ACE models (Carsault et al. [2018]). Hence, we analyze the interdependence of the representations of chords and their associated distances, the precision of the chord alphabets, and the architecture of our neural network models. Thus, we propose to use training losses based on musical theory for the multi-step chord sequence prediction. By performing an in-depth analysis of our results, we would like to uncover a set of related insights based on statistical models, and also formalize the musical meaning of some prediction errors.

[*]https://esling.github.io/atiam-ml-project

```
<chord>      ::=    <pitchname> ":" <shorthand> ["("<ilist>")"]["/"<interval>]
                  | <pitchname> ":" "("<ilist>")" ["/"<interval>]
                  | <pitchname> ["/"<interval>]
                  | "N"

<pitchname>  ::=  <natural> | <pitchname> <modifier>

<natural>    ::=  "A" | "B" | "C" | "D" | "E" | "F" | "G"

<modifier>   ::=  "b" | "#"

<ilist>      ::=  ["*"] <interval> ["," <ilist>]

<interval>   ::=  <degree> | <modifier> <interval>

<degree>     ::=  <digit> | <digit> <degree> | <degree> "0"

<digit>      ::=  "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

<shorthand>  ::=   "maj" | "min" | "dim" | "aug" | "maj7" | "min7" | "7"
                 | "dim7" | "hdim7" | "minmaj7" | "maj6" | "min6" | "9"
                 | "maj9" | "min9" | "sus2" | "sus4"
```

Figure 1: Syntax of Chord Notation in the Realbook dataset, image taken from Harte [2010]

2 Chord dataset

Exercice 1 - Exploiting the chord dataset Our different models will be trained on the *real book dataset* (Choi et al. [2016]) composed of 2847 tracks taken from the real book [2]. The dataset is available at this link : `https://github.com/keunwoochoi/lstm_real_book`.

The original chord alphabet uses a vocabulary of 1259 labels. This great diversity comes from the precision level in the chosen syntax (Figure 1). Starting with the assumption that the generation of coherent chord progressions only needs information on harmonic functions (without taking harmonic enrichments into account), we propose first to apply a clustering of the elements from the initial alphabet A_0 to obtain three different alphabets. They represents different hierarchical organization corresponding to the harmonization of the major scale using triads or tetrachords, which is often used to write chord progressions (the mapping will be provided) :

- A_1 : Major, minor, diminished: N (which means no chord), maj, min, dim;
- A_2 : Major, minor, seventh, diminished: N, maj, min, maj7, min7, 7, dim, dim7;
- A_3 : Major, minor, seventh, diminished, augmented, suspended: N, maj, min, maj7, min7, 7, dim, dim7, aug, sus;

1. Read the tutorial and create your custom chord sequence dataloaders `https://pytorch.org/tutorials/beginner/data_loading_tutorial.html`
2. Include different chord reduction of the initial alphabet in the dataloader, you can use existing code `https://github.com/carsault/MLSP19/blob/master/utilities/chordUtil.py`
3. Implement other chord aggregations that could be considered as "equivalent classes", for example by putting under the same label the usual harmonic substitutions (e.g. C:maj == A:min), or the substitutions that use the key and the harmonic degree of each chord in the sequence (e.g. vi == I). Indeed, our guess is that our system could benefit from this introduction of higher level of understanding instead of considering exhaustive chord qualities. (See Carsault et al. [2018])
4. Extract some information on the dataset (total number of chords for each chord class, chord repetition within chord sequences, chord transition matrix, ...)
5. (Bonus) Use the downbeat information to collect more features of the chord dataset

[2]The real book dataset is a compilation of jazz classics that has been firstly collected by the students of the Berklee College of Music during the seventies. As of today we count a lot of existing books (e.g Realbook (1, 2, 3), New Realbook (1, 2, 3), the Fakebook, the real Latin book)

3 Neural networks with Pytorch

In order to understand the different models that we will use for the chord sequence prediction, we will perform their implementation by relying on the *Pytorch* framework for Python.

Exercise 2: Learning PyTorch and useful libraries. This exercise includes learning basic programmation skills that you will need for the following work. The aim is to implement different model in order to evaluate them on our chord dataset

1. Install and learn PyTorch through basic tutorials `http://pytorch.org/tutorials/`
2. Create your chord generator model with the help of this tutorial on RNN/LSTM `http://colah.github.io/posts/2015-08-Understanding-LSTMs/` and this Pytorch tutorial `https://pytorch.org/tutorials/intermediate/char_rnn_generation_tutorial.html`
3. Implement a MLP that will take as input the same chord sequence representation (sequence of one hot vector) and output a sequence of probability vectors.
4. Train and evaluate your models on the chord dataset with different types of reduction.
5. Perform qualitative analysis with the help of the ACE analyzer `http://repmus.ircam.fr/dyci2/ace_analyzer`, that analyses musical relationships between chords.
6. (Bonus) Try to implement other models such as CNN or VAE and compare their performances with the LSTM and MLP.

Exercice 3 - Improving the cost function In the previous trained model, the cost function is ofter a categorical cross-entropy that doens't take into account the musical relatioship between the different chord classes. We want to upgrade this cost function with music theory considerations. Thus, we will modify it in order to train our model with other chord distances.

1. Study the implemented distances between chords `https://github.com/carsault/MLSP19/blob/master/utilities/distance.py`
2. Implement the distance between chords proposed in this article Paiement et al. [2005] (Paragraph 2. Representation) (The code for the chord to vector transformation will be provided)
3. Integrate this distance in the learning of our models and evaluate them quantitatively and qualitatively (Carsault et al. [2018]).
4. (Bonus) Implement another distance of your inspiration based on the music/perceptive/cognitive theory.

References

Jérôme Nika, Ken Déguernel, Axel Chemla, Emmanuel Vincent, and Gérard Assayag. Dyci2 agents: merging the" free"," reactive", and" scenario-based" music generation paradigms. In *Proceedings of ICMC*, 2017.

Tristan Carsault, Andrew McLeod, Philippe Esling, Jérôme Nika, Eita Nakamura, and Kazuyoshi Yoshii. Multi-step chord sequence prediction based on aggregated multi-scale encoder-decoder networks. In *MLSP*, 2019.

Tristan Carsault, Jérôme Nika, and Philippe Esling. Using musical relationships between chord labels in automatic chord extraction tasks. In *Proceedings of ISMIR*, 2018.

Keunwoo Choi, George Fazekas, and Mark Sandler. Text-based LSTM networks for automatic music composition. *arXiv preprint arXiv:1604.05358*, 2016.

Christopher Harte. *Towards automatic extraction of harmony information from music signals*. PhD thesis, 2010.

Jean-François Paiement, Douglas Eck, Samy Bengio, and David Barber. A graphical model for chord progressions embedded in a psychoacoustic space. In *Proceedings of the 22nd international conference on Machine learning*, pages 641–648. ACM, 2005.