Prospectus

Hugh Zabriskie

2 October 2015

1 Introduction

The year 1959 saw the first serious application of computers to the task of music composition with The *Illiac Suite* for string quartet. Written by Lejaren Hiller and Leonard Isaacson at the University of Illinois, Urbana-Champaign (and named for the ILLIAC supercomputer built on their campus), the suite was composed by generating a series of random values and using them to determine different properties of the work, such as pitch, duration, and dynamics. Since then, more sophisticated models have been developed to automate musical processes - in particular, for the task of automatic harmonization. This task is defined as follows: given a melody and a set of works as input, produce a suitable harmonic counterpoint to the melody, either in the form of harmonic representations (i.e. chord symbols) or as additional voices. Menzel et. al. (1992) developed an artificial neural network (ANN) called HARMONET that was trained on numerical representation of multiple chorales and able to learn *by example* how to produce its own chorales - harmonizations so accurate that they were judged "on the level of an improvising organist" [4]. And interestingly, one of the first undergraduate theses in Computer Science and Music was written by Chris Thorpe (A.B. '98), who used Markov chains to generate bass lines for melodies extracted from Bach's Chorales. [12]. Many others have explored the automation of this task across different corpuses, and the results have been astonishingly "correct".

However, the notable successes - as much as the evident failures - in automating musical processes have led some to question the utility and purpose of these algorithms. Neural networks are limited by the input that they are trained on. They are able to reproduce based on examples, but unable to be highly original in their compositions. Indeed, algorithmic composition is most successful when the harmonic and melodic space is well-defined by a series of training examples. But there *are* many instances where this context is well-defined, and therefore an important place exists for computers in the process of composition - given a complex but highly patterned musical process, a machine is able to reproduce it with exceptional speed and accuracy. Ultimately, these machines should be an aide to the composer with a well-defined purpose, rather than becoming a "compositional crutch" [6].

2 Goals and Questions

2.1 Objective

This paper will examine the application of recurrent neural networks to the task of automatic harmonization. The network will be applied to three sets of musical works.

1. The task of harmonizing the soprano line in Bach's Four-Voice Chorales with a bass line and inner voices.
2. Harmonic analysis of Bach's 6 Unaccompanied Cello Suites.
3. As a final consideration, harmonic analysis of jazz bebop solos.

2.2 Goals

The goal is to be a scientist, rather than merely model the process of harmonization. Experiments will be constructed such that either a successful or failed harmonization leads to interesting conclusions. In this spirit, the network will be trained on different corpuses with different feature representations to determine which features are most indicative of harmony, phrase structure, and so on. Ideally, the machine learns to output harmony a sense of *hierarchy* and that accelerates towards cadences.

3 Proposed Table of Contents

1. Introduction for Musicians
2. Introduction for Computer Scientists
3. Multi-Voice Harmonic Analysis
4. Harmonic Classification in Melodies
5. Application to Jazz
6. Appendix
    A. Musical Examples and Sources
    B. Code Listing

This proposed outline is intended to make the thesis readable for musicians and computer scientists alike. For the musician, the greatest challenge is to grasp the concept of a neural network and learn enough vocabulary to understand the statistical computations being applied. For the computer scientist, their challenge is to understand, in brief, the nature of tonal harmony. Given that both challenges are overcome, the reader will understand the conversion of a musical work into a numerical representation that can be fed into a neural net.

4 Methodology

4.1 Music21

The computation models presented in this paper will rely on the MUSIC21 Python library, developed by Professor Michael Scott Cuthbert at MIT. MUSIC21 provides a data structure called a *stream* that can represent a musical score, part, or measure in memory. Scores are input as MIDI files, which music21 parses into its own Python representation that can be altered for preprocessing before being converted into a numerical representation. Professor Cuthbert is also a non-faculty advisor for this thesis, and I will be able to leverage his expertise in music data representation to build flexible and efficient programs for processing large volumes of scores.

4.2 Torch and Odyssey

Torch is a scientific computing framework with wide support for machine learning algorithms. It is built on top of Lua, which is a powerful, but lightweight embeddable scripting language with a similar high-level focus as Python. I will use the HDF5 binary data format to transfer the preprocessed scores in Python over to Lua for training the neural network. To run models, I will execute scripts on the Harvard FAS Odyssey Computing Cluster. The account was provided by the Harvard Natural Language Processing research team, led by thesis advisor Prof. Sasha Rush.

4.3 Data Collection

All scores are initially stored as MIDI files. For this paper, I selected Bach's 371 harmonized chorales printed in the Riemenschneider collection, for which all of the scores are provided and indexed in the music21 corpus module. MIDI files for Bach's 6 Unaccompanied Cello Suites were found online at `http://www.jsbach.net/midi/midi_solo_cello.html`, which will be cross-checked with a known printed edition. And while MIDI files for jazz solo transcriptions are more sparse, there are an abundance of PDFs available and a number of programs for converting digital scores into MIDI and musicXML formats.

As an appendix, a listing of MIDI scores will be provided, as well as segments of code from the more crucial programs used to process scores and propagate the network.

References

[1] Alex Chilvers and Menno van Zaanen. Chorale harmonization in the style of js bach a machine learning approach. *Proceedings of the MML 2008 International Workshop on Machine Learning and Music held in conjunction with ICML/COLT/UAI 2008, Helsinki, Finland*, 2008.

In this report, the authors employ a feed-forward neural network from the SNNS 4.2 package to attempt harmonize the 4-voice Bach Chorale. They experiment with different data representations for the chorale to determine the most effective one. They had little success with training the neural net to recognize good chord pairs, but later saw a significant increase in success training it on chord prediction. The report by Chilvers and van Zaanen is by far the most ”scientific”, in that the authors experiment with different input features and tasks and at each step evaluate the reasons for success or failure. However, this report was evidently written by non-musicians, and their approaches to the task of harmonization demonstrate only a partial understanding of tonal harmony.

[2] Douglas Eck and Juergen Schmidhuber. A first look at music composition using lstm recurrent neural networks. *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*, 2002.

Upon examining attempts at automated composition using standard recurrent neural networks (RNNs), the authors Eck and Schmidhuber conclude that these models alone fail to grasp larger musical structures, and therefore are unsuccessful at generating convincing musical compositions. However, a subset of RNNs labeled LSTMs (Long Short-Term Memory) have shown much greater promise due to their ability to retain information about decisions in previous time frames. With respect to music generation, the authors demonstrate its ability to successfully learn a 12-bar blues form and improvise melodies over those harmonies.

[3] Douglas Eck and Jurgen Schmidhuber. Finding temporal structure in music: Blues improvisation with lstm recurrent networks. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 747–756. IEEE, 2002.

This article is a further development of the ideas Eck proposes in his other 2002 paper (cite-key: eck2002structure), where he demonstrates the ability of LSTMs to develop an understanding of larger musical structures, such as the 12-bar blues form. Here, Eck argues that LSTMs are also highly effective at learning to compose music by analyzing examples of a machine that is able to produce blues melodies. Eck is one of the few researchers to explore the realm of computational harmonic analysis in jazz, and he loves many topics untouched where others could explore.

[4] Hermann Hild, Johannes Feulner, and Wolfram Menzel. Harmonet: A neural net for harmonizing chorales in the style of js bach. In *Advances in Neural Information Processing Systems*, pages 267–274, 1992.

HARMONET is a neural network designed to harmonized a chorale melody, taking as the soprano part. The authors proposed a multi-step process, using different classifiers to compose a bass line, to compose inner voices and to insert eighth note to ehance voice leading. Its results have been impressive, and the author claims that HARMONET's output is comparable to an "improvising organist."

[5] Dominik Hörnel and Wolfram Menzel. Learning musical structure and style with neural networks. *Computer Music Journal*, 22(4):pp. 44–62, 1998.

Building on the HARMONET program they implemented six years before, Menzel et. al. analyze the potential for neural networks to adapt to new corpuses and learn musical strucutre. They provide additional mathematical background on HARMONET, which will be useful for implementing new RNNs. They also try to define basic (if not simplistic) formulas for attributes like accuracy of style (HSV, or Harmonic Style Value).

[6] Bruce L. Jacob. Algorithmic composition as a model of creativity. *Organised Sound*, 1(3):157–165, 1996.

Bruce's article provides context on the debate over the automation of musical processes. Fundamentally, he argues for the use of algorithmic composition as a "compositional tool", but somewhat subjectively states that it is "often considered a cheat" when the composer feels uninspired. He asks who is responsible for the music produced, and then questions how a composer's increasing ability to reproduce a compositional process is affecting the value of creativity. Notably, this article was written by a member of the Univeristy of Michigan's Advanced Computer Architecture Laboratory, rather than a musician. It nevertheless provides a critical perspective on the role of automatic composition, which remains a hot debate today among musicians - and evidently computer scientists, too.

[7] Martin Elmer Jorgensen and Soren Tjagvad Madsen. Harmonisation of bach chorales: Kbs project report. Technical report, University of Aarhus, August 2002.

This report represents a fairly standard approach to the task of automatic harmonization of Bach chorales. Given the melody of a chorale, the authors developed a neural network to provide harmonic support for each note in the melody, which has been abstracted to series a quarter notes for the sake of evenly distributed time frames. Input features include past and previous elements of the melody, previously determined harmonies, and contextual elements of the current soprano note, such as beat strength. After experimenting with several different data representations for the feature vectors, they discovered a relatively effective solution when they represented the output vector as a range of MIDI notes from which the alto, tenor, and bass voices were selected. This paper is also somewhat dated and doesn't provide an adequate explanation of their neural network construction. However, their method is preprocessing (that is, abstracting chorales into quarter-note time frames) is intruiging and likely to be used in my initial experiments with RNNs.

[8] Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. Blog, May 2015.

Karpathy's popular blog post is meant as an introduction to recurrent neural networks for an audience familiar with machine learning vocabulary but interested in a less technical overview. He explains the fundamental advance made with RNNs over "vanilla" neural networks: they are able to remember recent examples. As a result, these networks are able to make decisions with an intuition for hierarchy and structure in its ouput. He provides some undoubtedly mind-boggling examples of the power of RNNs, demonstrating their ability to essentially learn valid English, LaTeX, or C syntax while also providing phrase structure.

[9] Dylan Jeremy Nagler. Schubot: Machine learning tools for the automated analysis of schubertâs lieder. Honors thesis, Harvard University, March 2014.

Dylan Nagler's undergraduate represents one of the most recent attempts to apply a variety of machine learning models to musical tasks. This demonstrates how the basic questions about automatic harmonization continue to be asked many years later, although the author here applies them to a less common data set - Schubert lieder. He first describes a number of supervised learning methods which might be applied, including hidden Markov models and PCFGs (grammars). He then describes a program he wrote to harmonically analyze Schubert lieder and the metholodogy behind his approach. Nagler is very descriptive in explaining his methodology, which will be additionally useful because of his use of the music21 musicology library to preprocess his data.

[10] Chris Olah. Understanding lstm networks, August 2015.

In his recent blog post, Chris Olah describe a special model of recurrent neural networks known as LSTM Networks, or Long Short-Term Memory Networks. He describes the significant advances that recurrent neural networks made over "vanilla" (or non-recurrent) neural networks by allowing information to persist within the model. Theoretically, RNNs should be able to draw on information from previous decisions to make its current one, but in practice they have only proven effective when the distance between the relevant past information and the time frame in which it is required is small. LSTMs solved this problem by altering the structure of the network's hidden activation layers to create a pipeline of persisting information and a series of control gates for allowing new information to enter the pipeline. This blog post is aimed at machine learning researching with a general understanding of neural networks and are seeking a more visual and approachable introduction to LSTMs. While a young author, Chris is a highly accomplished researcher and engineer in the field, and his explanations are concise, descriptive, and well-researched, citing several relevant papers.

[11] George Papadopoulos and Geraint Wiggins. Ai methods for algorithmic composition: A survey, a critical view and future prospects. *AISB Symposium on Musical Creativity*, pages 110–117, 1999.

This paper examines a variety of methods for algorithmic composition. For each method (i.e. systems that learn, grammars, etc.), the major attempts using this methodology are examined, and then the authors provide a list of advantages and disvantages to exploring the approach further. They mention several past approaches taken to the task of harmonization - as a satisfaction constraint problem, as a generative grammar, or as a learn-by-example approach using machine learning. This paper was written at the turn of the century and therefore lacks some of the more recent research on harmonization using sophisticated neural networks. However, it remains useful as a well-sourced appendix of attempts at automatic harmonization using a variety of computational models. The extensive bibliography will also come in handy for future research.

[12] Christopher A. Thorpe. C.p.u. bach : using markov models for chorale harmonization. Honors thesis, Harvard University, March 1998.

Unlike many past and future attempts, Chris Thorpe's A.B. thesis explores the task of harmonizing Bach chorales with a bass line rather than a complete set of voices. He uses Markov chains to model the harmonization task, representating a harmony at time T as a given "state" that can transition to any one of a set of other harmonies, in each case with a pre-determined probability of that transition occurring. Thorpe models harmonization in this form as a "completive" rather than generative task, in that the produced bass line is one of a set of valid options. This paper is one of the most notable Harvard undergraduate theses in Computer Science and Music. Thorpe is forthcoming about his methodology and the reasoning behind it, while also acknowledging the limitations of a Markov-based approach.

[13] Petri Toiviainen. Modeling the target-note technique of bebop-style jazz improvisation: An artificial neural network approach. *Music Perception: An Interdisciplinary Journal*, 12(4):pp. 399–413, 1995.

Petri Toivainen presents an artificial neural network (ANN) that given a series of chord changes is able to improvise over them in a jazz bebop style. The target-note technique describes provides signposts at the beginning of each chord change, giving the ANN targets with which to arrive at using a series of melodic patterns. The result is a suprisingly decent melodic improvisation given a small training set - consisting of Clifford Brown's solos on "All the Things You Are" and "Getrude's Bounce".Toivainen's paper is an early example of a computational approach to jazz composition.