Judah Daniels

Inferring Harmony from Free Polyphony

Computer Science Tripos – Part II

Clare College

July, 2023

Declaration of originality

I, Judah Daniels of Clare College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose. I am content for my dissertation to be made available to the students and staff of the University.

Signed Judah Daniels

Date March 23, 2023

Proforma

Candidate Number:	**2200D**
Project Title:	**Inferring Harmony from Free Polyphony**
Examination:	**Computer Science Tripos – Part II, July, 2023**
Word Count:	**1587**[1]
Code Line Count:	**2006**
Project Originator:	**Christoph Finkensiep**
Supervisor:	**Dr Peter Harrison**

Original Aims of the Project

Work Completed

All that has been completed appears in this dissertation.

Special Difficulties

None

[1]This word count was computed by `detex diss.tex | tr -cd '0-9A-Za-z \n' | wc -w`

Contents

List of Figures

Acknowledgements

Chapter 1

Introduction

This dissertation explores ... We present .. which extends a recent model, addressing problems of intractability by using Heuristic search methods. We show that our novel heuristic search method ...

1.1 Motivation

1.2 Related Work

1.3 Aims

Chapter 2

Preparation

In this chapter, I present the work which was undertaken before the code was written. After a brief description of my starting point, I provide an exposition of the Proto-voice Model which forms the foundation of this project. Subsequently, I discuss probabilistic programming and Bayesian inference, including a probabilistic model of harmony. Finally, I describe the software engineering techniques and principles used throughout the project.

2.1 Starting Point

2.1.1 Relevant courses and experience

2.1.2 Existing codebase

2.2 The Protovoice Model

2.3 Probabilistic Programming

2.3.1 Bayesian Inference

2.3.2 Probabilistic Model of Harmony

2.4 Heuristic Search Algorithms

2.5 Requirements Analysis

2.5.1 Main deliverables

2.5.2 Dependency Analysis

2.6 Software Engineering Techniques

Justified and documented selection of suitable tools; good engineering approach.

Table 2.1: Languages, libraries and tools

Tool	Purpose	License
Haskell	Main language	...
GHC	Compiling and profiling to inspect time performance and memory usage	GPL-3.0+
Haskell-Musicology	...	...
Dimcat	...	...
Python	...	...
Numpy	...	...
Pandas	...	...
MS3	...	...
Musescore 3	...	...
Protovoice Annotation Tool	...	...
Git	Version Control, Continuous Integration	...

2.6.1 Development model

Include Gantt chart.

2.6.2 Languages, libraries and tools

The chapter will also cite any new programming languages and systems which had to be learnt

Chapter 3

Implementation

3.1 Repository Overview:

The following describes an overview of the project repository:

3.2 Core Implementation

3.3 Baseline implementation

3.4 Random Sample Parser

3.5 Random Choice Search

3.6 Extension Implementation

3.6.1 Probabilistic Model of Harmony

3.6.2 Heuristic Design

3.6.3 Heuristic Search

3.7 Testing

Table 3.1: Repository Overview

File/Folder	Description	LOC
protovoices-haskell/	Root directory	1972
├─ src/		
│ ├─ HeuristicParser.hs, HeuristicSearch.hs	Core Implementation (Section x)	470
│ ├─ RandomChoiceSearch.hs, RandomSampleParser.hs	Baseline Implemetation (Section x)	121
│ ├─ Heuristics.hs, PBHModel.hs	Extension Implementation (Section x)	383
│ ├─ FileHandling.hs	Utilities	188
│ └─ ...		
├─ app/		
│ └─ MainFullParse.hs	Entry Point	121
├─ harmonic-inference		
├─ experiments/		115
│ ├─ preprocess.ipynb		
│ ├─ dcml_params.json		
│ └─ inputs/		
└─ test/	Unit Tests (Section x)	611

Chapter 4

Evaluation

In this chapter, I provide qualitative and quantitative evaluations of the work completed. I then provide and interpret evidence to show that the success criteria were met.

The main questions to answer are as follows:

- *Can the proto-voice model be used to accurately infer chord labels?*
- *Can the proto-voice model be used to practically infer chord labels?*
- *How well my heuristic search algorithms infer chord labels?*

4.1 Accuracy

4.2 Performance

4.3 Heuristic Search (Extension)

4.4 Success Criteria

4.5 Limitations

Chapter 5

Conclusions

In this chapter, I first discuss the success achieved by the project then offer a reflection on lessons learned. Finally, I consider the directions in which there is potential for future work.

5.1 Achievements

5.2 Lessons learned

5.3 Future Work

[?]

Bibliography

[1] Harmonic Syntax in Time: Rhythm Improves Grammatical Models of Harmony. *Proceedings of the 20th ISMIR Conference* (2019).

[2] Chen, T.-P., and Su, L. Functional Harmony Recognition of Symbolic Music Data with Multi-task Recurrent Neural Networks, Sept. 2018.

[3] Chen, T.-P., and Su, L. Harmony Transformer: Incorporating Chord Segmentation into Harmony Recognition. In *International Society for Music Information Retrieval Conference* (2019).

[4] Cohen, D. E. "The Imperfect Seeks Its Perfection": Harmonic Progression, Directed Motion, and Aristotelian Physics. *Music Theory Spectrum 23*, 2 (Oct. 2001), 139–169.

[5] Feisthauer, L., Bigo, L., Giraud, M., and Levé, F. Estimating keys and modulations in musical pieces. In *Sound and Music Computing Conference (SMC 2020)* (June 2020).

[6] Finkensiep, C. *The Structure of Free Polyphony.* PhD thesis, EPFL, Lausanne, 2023.

[7] Finkensiep, C., and Rohrmeier, M. MODELING AND INFERRING PROTO-VOICE STRUCTURE IN FREE POLYPHONY.

[8] Gjerdingen, R. Cognitive Foundations of Musical Pitch Carol L. Krumhansl. *Music Perception: An Interdisciplinary Journal 9* (July 1992), 476–492.

[9] Goodman, J. Semiring Parsing. *Computational Linguistics 25*, 4 (1999), 573–606.

[10] Gotham, M., Kleinertz, R., Weiss, C., Müller, M., and Klauk, S. What if the 'When' Implies the 'What'?: Human harmonic analysis datasets clarify the relative role of the separate steps in automatic tonal analysis. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021* (2021), J. H. Lee, A. Lerch, Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., pp. 229–236.

[11] Granroth-Wilding, M. Harmonic Analysis of Music Using Combinatory Categorial Grammar. *The University Of Edinburgh* (2013).

[12] Humphrey, E. J., and Bello, J. P. Four Timely Insights on Automatic Chord Estimation.

[13] Johannes, H. Ms3 - Parsing MuseScore 3. https://github.com/johentsch/ms3, 2021.

[14] Kidney, D. O., and Wu, N. Algebras for weighted search. *Proceedings of the ACM on Programming Languages 5*, ICFP (Aug. 2021), 1–30.

[15] Koops, H. V., de Haas, W. B., Burgoyne, J. A., Bransen, J., Kent-Muller, A., and Volk, A. Annotator subjectivity in harmony annotations of popular music. *Journal of New Music Research 48*, 3 (May 2019), 232–252.

[16] Korzeniowski, F., and Widmer, G. Improved Chord Recognition by Combining Duration and Harmonic Language Models, Aug. 2018.

[17] Krumhansl, C. L., and Kessler, E. J. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review 89* (1982), 334–368.

[18] Lerdahl, F., and Jackendoff, R. *A Generative Theory of Tonal Music*, repr. ed. MIT Press, Cambridge, Mass., 2010.

[19] Marsden, A. Schenkerian Analysis by Computer: A Proof of Concept. *Journal of New Music Research 39*, 3 (Sept. 2010), 269–289.

[20] Mauch, M., Mullensiefen, D., Dixon, S., and Wiggins, G. Can Statistical Language Models be used for the Analysis of Harmonic Progressions?

[21] McLeod, A., and Rohrmeier, M. A Modular System for the Harmonic Analysis of Musical Scores using a Large Vocabulary. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021* (2021), J. H. Lee, A. Lerch, Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., pp. 435–442.

[22] Neuwirth, M., Harasim, D., Moss, F. C., and Rohrmeier, M. The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets. *Frontiers in Digital Humanities 5* (July 2018), 16.

[23] Ni, Y., Mcvicar, M., Santos-Rodriguez, R., and De Bie, T. An end-to-end machine learning system for harmonic analysis of music, July 2011.

[24] Pardo, B., and Birmingham, W. P. Algorithms for Chordal Analysis. *Computer Music Journal 26*, 2 (June 2002), 27–49.

[25] Pickens, J. *Harmony Modeling for Polyphonic Music Retrieval.* PhD thesis, University of Massachusetts, 2004.

[26] Quinn, I. Are Pitch-Class Profiles Really "Key for Key"? *Zeitschrift der Gesellschaft für Musiktheorie [Journal of the German-Speaking Society of Music Theory] 7* (Jan. 2010).

[27] Rocher, T., Robine, M., Hanna, P., and Strandh, R. Dynamic Chord Analysis for Symbolic Music.

[28] Sapp, C. Visual hierarchical key analysis. *Computers in Entertainment 3* (Oct. 2005), 1–19.

[29] Sidorov, K., Jones, A., and Marshall, D. MUSIC ANALYSIS AS A SMALLEST GRAMMAR PROBLEM.

[30] Temperley, D. A Bayesian Approach to Key-Finding. In *Music and Artificial Intelligence*, G. Goos, J. Hartmanis, J. van Leeuwen, C. Anagnostopoulou, M. Ferrand, and A. Smaill, Eds., vol. 2445. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 195–206.

[31] Temperley, D. A Bayesian Approach to Key-Finding. In *Music and Artificial Intelligence* (Berlin, Heidelberg, 2002), C. Anagnostopoulou, M. Ferrand, and A. Smaill, Eds., Lecture Notes in Computer Science, Springer, pp. 195–206.

[32] Temperley, D. A Unified Probabilistic Model for Polyphonic Music Analysis. *Journal of New Music Research 38*, 1 (Mar. 2009), 3–18.

Appendix A

Additional Information

Appendix B

Project Proposal

Inferring Harmony from Free Polyphony

Judah Daniels

March 23, 2023

DOS: Prof. Larry Paulson

Crsid: jasd6
College: Clare College

B.1 Abstract

A piece of music can be described using a sequence of chords, representing a higher level harmonic structure of a piece. There is a small, finite set of chord types, but each chord can be realised on the musical surface in a practically infinite number of ways. Given a score, we wish to infer the underlying chord types.

The paper *Modeling and Inferring Proto-voice Structure in Free Polyphony* describes a generative model that encodes the recursive and hierarchical dependencies between notes, giving rise to a grammar-like hierarchical system [**?**]. This proto-voice model can be used to reduce a piece into a hierarchical structure which encodes an understanding of the tonal/harmonic relations of a piece.

Christoph Finkensiep suggests in his paper that the proto-voice model may be an effective way to infer higher level latent entities, such as harmonies or voice leading schemata. Thus in this project I will ask the question: is this parsing model an effective way to annotate harmonies? By 'effective' we are referring to two things:

- Accuracy: can the model successfully emulate how experts annotate harmonic progressions in musical passages?
- Practicality: can the model be used to do this within a reasonable time frame?

While the original model could in theory be used to generate harmonic annotations, its exhaustive search strategy would be prohibitively time-consuming in practice for any but the shortest musical extracts; one half measure can have over 100,000 valid derivations [**?**]. My approach will be to explore the use of heuristic search algorithms to solve this problem.

B.2 Substance and Structure

B.2.1 Core: Search

The core of this project is essentially a search problem characterised as follows:

- The state space S is the set of all possible partial reductions of a piece along with each reduction step that has been done so far.
- We have an initial state $s_o \in S$, which is the empty reduction, corresponding to the unreduced surface of the piece. The score is represented as a sequence of slices grouping notes that sound simultaneously. We are also given the segmentation of the original chord labels that we wish to retrieve.
- We have a set of actions, A modelled by a function $action : A \times S \rightarrow S$. These actions correspond to a single reduction step.
  - The reduction steps are the inverses of the operations defined by the generative proto-voice model.
- Finally we have a goal test, $goal : S \rightarrow \{true, false\}$ which is true iff the partial reduction s has exactly one slice per segment of the input.

  - This means the partial reduction s contains a sequence of slices which start and end positions corresponding to the segmentation of the piece.

- At the first stage, this will be implemented using a random graph search algorithm, picking each action randomly, according to precomputed distributions.

B.2.2 Core: Evaluation

The second core task is to create an evaluation module that iterates over the test dataset, and evaluates the partial reduction computed by the search algorithm above. This will be done by comparing the outputs to ground truth annotations from the Annotated Beethoven Corpus.

In order to do this I will make use of the statistical harmony model from Finkensiep's thesis, *The Structure of Free Polyphony* [**?**]. This model provides a way of mapping between the slices that the algorithm generates and the chords in the ground truth. This can be used to empirically measure how closely the slices match the expert annotations.

B.2.3 Extension

Once the base search implementation and evaluation module have been completed, the search problem will be tackled by heuristic search methods, with different heuristics to be trialled and evaluated against each other. The heuristics will make use of the chord profiles from Finkensiep's statistical harmony model discussed above. These profiles relate note choices to the underlying harmony. Hence the heuristics may include:

- How the chord types relate to the pitches used.
- How the chord types relate which notes are used as ornamentation, and the degree of ornamentation.
- Contextual information about neighboring slices

B.2.4 Overview

The main work packages are as follows:

Preliminary Reading – Familiarise myself with the proto-voice model, and read up on similar models and their implementations. Study heuristic search algorithms.

Dataset Preparation – Pre-process the Annotated Beethoven Corpus into a suitable representation for my algorithm.

Basic Search – Implement a basic random search algorithm that takes in surface and segmentations, and outputting the sequence of slices matching the segmentations.

Evaluation Module – Implement an evaluation module to evaluate the output from the search algorithm.

End-to-end pipeline – Implement a full pipeline from the data to the evaluation that can be used to compare different reductions.

Heuristic Design – Extension – Trial different heuristics and evaluate their performance against each other.

Dissertation – I intend to work on the dissertation throughout the duration of the project. I will then focus on completing and polishing the project upon completion.

B.3 Starting Point

The following describes existing code and languages that will be used for this project:

Haskell – I will be using Haskell for this project as it is used in the proto-voice implementation. It must be noted that my experience with Haskell is limited, as I was first introduced to it via an internship this summer (July to August 2022).

Python – Python will be used for data handling. I have experience coding in Python.

Prior Research - Over the summer I have been reading the literature on computational models of music, as well as various parsing algorithms such as semi-ring parsing [**?**], and the CYK algorithm, which is used in the implementation of the proto-voice model.

Protovoices-Haskell – The paper *Modeling and Inferring Proto-Voice Structure in Free Polyphony* [**?**] includes an implementation of the proto-voice model in Haskell. A fork of this repository will form the basis of my project. This repository includes as parsing module which will be used to perform the actions in the search space of partial reductions. There is module that can exhaustively enumerate reductions of a piece, but this is infeasible in practice due to the blowup of the derivation forest.

MS3 – This is a library for parsing MuseScore Files and manipulating labels [**?**], which I will use as part of the data processing pipeline.

ABC – The *Annotated Beethoven Corpus* [**?**] contains analyses of all Beethoven string quartets composed between 1800 and 1826), encoded in a human and machine readable format. This will be used as a dataset for this project.

B.4 Success Criteria

This project will be deemed a success if I complete the following tasks:

- Develop a baseline search algorithm that uses the proto-voice model to output a partial reduction of a piece of music up to the chord labels.
- Create an evaluation module that can take the output of the search algorithm and quantitatively evaluate its accuracy against the ground truth annotations by providing a score based on a statistical harmony model.
- Extension: Develop one or more search algorithms that use additional heuristics to inform the search, and compare the accuracy with the baseline algorithm.

B.5 Timetable

Time frame	Work	Evidence
Michaelmas (Oct 4 to Dec 2)		
Oct 14 to Oct 24	*Oct 14*: Final proposal deadline. Preparation work: familiarise myself with the dataset and the proto-voice model implementation. Work on manipulating reductions using the proto-voice parser provided by the paper.	None
Oct 24 to Nov 7	Dataset preparation and handling.	Plot useful metrics about the dataset using Haskell
Nov 7 to Nov 21	Random Search implementation	None
Nov 21 to Dec 5	Evaluation Module. Continue with search implementation.	Evaluate a manually created derivation and plot results
Vacation (Dec 3 to Jan 16)		
Dec 5 to Dec 11	Evaluate performance of random search. Begin to work on extensions	Plot results
Dec 10 to Dec 21	Trial different heuristics. Implement an end-to-end pipeline from input to evaluation.	None
Dec 21 to Dec 27	None	None
Dec 27 to Jan 10	Continue trialing and evaluating heuristics	*Fulfill success criterion: At least one heuristic technique gives better performance than random search.*
Lent (Jan 17 to Mar 17)		
Jan 4 to Jan 20	Buffer Period to help keep on track	None
Jan 20 to Feb 3	*Feb 3*: Progress Report Deadline. Write progress report and prepare presentation. Write draft *Evaluation* chapter	Progress Report (approx. 1 page)
Feb 3 to Feb 17	Prepare presentation.	*Feb 8 – 15*: Progress Report presentation
Feb 17 to Mar 3	*Feb 17*: How to write a Dissertation briefing. Write draft Introduction and Preparation chapters. Incorporate feedback on Evaluation chapter.	Send draft Introduction and Preparation chapter to supervisor

B.6 Resources

I plan to use my own laptop for development: MacBook Pro 16-inch, M1 Max, 32GB Ram, 1TB SSD, 24-core GPU.

All code will be stored on a GitHub repository, which will guarantee protection from data loss. I will easily be able to switch to using university provided computers upon hardware/software failure.

The project will be built upon work that has been done in the DCML (Digital cognitive musicology lab) based in EPFL. The files are in their Github repository, and I have been granted permission to access their in-house datasets of score annotations, as well as software packages which are used to handle the data.

B.7 Supervisor Information

Peter Harrison, head of Centre for Music and Science at Cambridge, has agreed to supervise me for this. We have agreed on a timetable for supervisions for this year. I am also working with Christoph Finkensiep, a PHD student at the DCML, and originator of the proto-voice model. Professor Larry Paulson has agreed to be the representative university teaching officer.