Addressing ambiguity in supervised machine learning: A case study on automatic chord labelling

Yaolong Ju

Music Technology Area
Department of Music Research
Schulich School of Music
McGill University, Montreal

April 2021

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy

Abstract

Chord labelling is an important analytical tool in Western tonal music with many possible applications. Manual chord labelling is a time-consuming process, and automatic chord labelling can be a promising alternative. However, most automated approaches to date have not sufficiently considered ambiguity in chord labelling, where multiple answers are possible due to different labelling strategies. In this dissertation, I present three ways of addressing this ambiguity in automatic chord labelling, using J. S. Bach's chorales as a case study.

First, I present a rule-based model suitable for generating preliminary chord labels for Bach chorales according to a specific analytical strategy, where each chord is assigned only a single label. These labels were later checked by an expert who provided corrections when necessary. Compared to generating annotations from scratch, the amount of work required for the expert was reduced, and the resulting labels combine the consistency of the rule-based model with the nuance of manual corrections. These annotations were then used to train machine learning models for automatic chord labelling.

Next, I created the Bach Chorales Figured Bass (BCFB) dataset that contains 139 chorales with both the original music and figured bass annotations. I then implemented four new rule-based algorithms for automatically generating chord labels for Bach chorales based on both figured bass annotations and the musical surface. Each algorithm is based on a different labelling strategy, which was applied to the dataset. The results contain multiple parallel labels for each chord and are presented as the new Bach Chorales Multiple Chord Labels (BCMCL) dataset.

Finally, I used the BCMCL dataset to explore multi-label learning and label distribution learning, two supervised machine learning paradigms that enable automatic chord labellers to generate multiple parallel answers for each chord, either in the form of binary labels or a distribution of probabilities, respectively. The parallel chord labels from BCMCL can also serve as a basis for exploring alternate evaluation metrics for automatic chord labellers in general. The resulting automatic chord labellers of this research can generate either single chord labels according to a specific analytical strategy or can generate multiple labels for a single chord based on a variety of different analytical perspectives.

Résumé

L'identification d'accord [*chord labelling*] est un outil analytique important utilisé en musique tonale occidentale et dont les applications sont nombreuses. Effectuer cette tâche à la main est un processus long et coûteux, et son automatisation semble une alternative prometteuse. Cependant, aucune des approches automatiques développées jusqu'à présent ne s'est penchée sur l'ambiguïté de cette identification, laquelle peut mener à plusieurs réponses possibles selon la stratégie employée. Cette thèse propose trois façons de résoudre cette ambiguïté en utilisant les chorals de J. S. Bach comme sujet d'étude.

Tout d'abord, nous proposons un modèle basé sur des règles qui génère une identification préliminaire d'accords à partir des chorals de Bach, et dont la stratégie analytique spécifique se base sur l'attribution d'un identifiant unique pour chaque accord. Les résultats ont été vérifiés par un spécialiste qui a fourni des corrections lorsque nécessaire. Comparée au processus intégralement manuel, la quantité de travail du spécialiste a été réduite; les identifiants finaux combinent la cohérence du modèle basé sur des règles et la nuance des corrections manuelles. Ils ont ensuite été utilisés pour entraîner des modèles d'apprentissage d'identification automatique des accords.

Ensuite, nous avons créé la BCFB (*Bach Chorales Figured Bass*), un ensemble de données de 139 chorals avec leur orchestration et leur basse chiffrée originales. Nous avons implémenté quatre nouveaux algorithmes basés sur des règles, lesquels génèrent automatiquement une identification des accords des chorals à partir de leur basse chiffrée et de leur contenu musical. Chaque algorithme est basé sur une stratégie d'identification différente appliquée à l'ensemble de données. Les résultats débouchent sur de nombreux identifiants parallèles pour chaque accord et sont présentés dans le nouvel ensemble de données BCMCL (*Bach Chorales Multiple Chord Labels*).

Enfin, nous avons utilisé l'ensemble de données BCMCL pour explorer l'apprentissage automatique d'identifiants multiples et l'apprentissage d'attribution d'identifiants. Ces deux paradigmes d'apprentissage supervisé permettent aux identificateurs automatiques de générer plusieurs réponses parallèles pour chaque accord, respectivement sous la forme d'identifiants uniques non superposés et de distribution de probabilités. Les identifiants parallèles d'accords hébergés par BCMCL peuvent également servir de base à d'autres métriques d'évaluation qui

pourront être appliquées aux identificateurs automatiques d'accords en général. Les identificateurs automatiques d'accords issus de notre recherche peuvent générer soit un identifiant unique par accord selon une stratégie analytique spécifique ou plusieurs identifiants parallèles par accord selon un panel de perspectives analytiques différentes.

Acknowledgements

First and foremost, I would like to thank my supervisors Professor Ichiro Fujinaga and Professor Cory McKay for all the generous support, including their invaluable research ideas, insights, and countless edits that helped make this dissertation possible. Their proficiency and rigorous training on critical thinking, academic writing, and career planning really shaped the researcher I am today. I would also like to thank my co-authors on the publications relating to this research: Samuel Howes and Sylvain Margot for their generous help on creating the datasets, offering manual annotations, and providing useful feedback on the music-theoretical and musicological aspects of this research; Nathaniel Condit-Schultz for his proofreading and implementing the rule-based algorithm in Section 5.1, and for the insightful discussions on chord labelling ambiguity, which eventually became the central focus of this dissertation; Luke Dahn for providing invaluable information on Bach chorales; and Jorge Calvo-Zaragoza for his great consultations on the design of machine learning models, and for his experience on how to excel as a Ph.D. student.

I am grateful for all the courses I took at McGill University, especially the Theory and Analysis I–IV courses. I would like to thank the instructors: Professors Edward Klorman, Christoph Neidhöfer, William Caplin, and Nicole Biamonte for their dedicated teaching and meaningful discussions, particularly on ambiguity in chord labelling and how this knowledge can be applied to the task of automatic chord labelling.

I would like to thank the *Centre for Interdisciplinary Research in Music Media and Technology* (CIRMMT) for providing enormous support for my Ph.D. studies, including travel awards, research funding, and various opportunities to cultivate interdisciplinary music research. Additionally, this research would not have been possible without the generous financial support from *Social Sciences and Humanities Research Council of Canada* (SSHRC) and *Fonds de Recherche du Québec-Société et Culture* (FRQSC).

I am grateful for my mentors and colleagues at Schulich School of Music at McGill University: Professors Jonathan Wild and Peter Schubert for supervising my research projects for the CIRMMT Student Awards; Professor Julie Cumming for offering many invaluable suggestions on my research and during the run-throughs of my paper presentations; Samuel Howes and Sylvain

Margot for being the collaborators on Interdisciplinary Research Project Seed Grant and CIRMMT Agile Seed Funding projects, respectively; Gabriel Vigliensoni for introducing me to the lab culture and helping me adapt to Ph.D. studies; Emily Hopkins for her extensive help on timesheets and expense reports, and for proof-reading my paper drafts; Martha Thomae for all the memorable conversations and chit-chat after work; Alex Daigle for his generous help on setting up virtual machines and deploying my research projects on the server; Gustavo Polins Pedro for being a considerate collaborator on the SIMSSA Database project, whom I enjoyed working with over the course of three years. I am also thankful for the Computational Tonal Studies Group: Jacob deGroot-Maggetti, Timothy Raja de Reuse, Laurent Feisthauer, Samuel Howes, Suzuka Kokubu, Sylvain Margot, Néstor Nápoles López, and Finn Upham for many valuable conversations and insightful feedback on my research. Also, I would like to thank Song Wang, Lingxiao Yang, Lei Fu, Behrad Madahi, and Wen Xiao for exchanging experiences on living abroad as international students.

Furthermore, I am grateful for my friends: Romeo Gagnon, Paul Alwin, Gabriel Remy-Handfield, Phani Pramod, Marie-Eve Fortier, Krishna Teja, and Nitin Reddy for the community support, great hangouts, culture exchanges, and national and international trips. You all made my living in Montreal a wonderful and memorable experience. Great thanks to Sylvain Margot and Gabriel Remy-Handfield, who offered invaluable help on the French translation of the abstract of this dissertation.

Finally, I would like to thank my family: Caifeng Pan, Jianhua Ju, and Yicheng Ju for their unconditional love and support. You are the reason why I am here today, with the privilege of studying abroad and finishing my Ph.D. studies.

Preface

This dissertation contains the candidate's original research and contributions, except for the information that is commonly accepted and references to other people's published work. This dissertation is organized as a monograph, which is presented in eight chapters and contains some content that has been previously published in peer-reviewed papers at academic conferences on music encoding, musicology, and music information retrieval:

- Chapter 5: Ju, Yaolong, Nathaniel Condit-Schultz, Claire Arthur, and Ichiro Fujinaga. 2017. "Non-Chord Tone Identification Using Deep Neural Networks." In *Proceedings of the 4th International Workshop on Digital Libraries for Musicology*, 13–16.
- Chapter 5: Condit-Schultz, Nathaniel, Yaolong Ju, and Ichiro Fujinaga. 2018. "A Flexible Approach to Automated Harmonic Analysis: Multiple Annotations of Chorales by Bach and Prætorius." In *Proceeding of the 19th International Society of Music Information Retrieval Conference*, 66–73.
- Chapter 5: Ju, Yaolong, Samuel Howes, Cory McKay, Nathaniel Condit-Schultz, Jorge Calvo-Zaragoza, and Ichiro Fujinaga. 2019. "An Interactive Workflow for Generating Chord Labels for Homorhythmic Music in Symbolic Formats." In *Proceeding of the 20th International Society for Music Information Retrieval Conference*, 862–69.
- Chapter 6: Ju, Yaolong, Sylvain Margot, Cory McKay, Luke Dahn, and Ichiro Fujinaga. 2020. "Automatic Figured Bass Annotation Using the New Bach Chorales Figured Bass Dataset." In *Proceeding of the 21st International Society for Music Information Retrieval Conference*, 640–46.
- Chapter 6: Ju, Yaolong, Sylvain Margot, Cory McKay, and Ichiro Fujinaga. 2020. "Automatic Chord Labelling: A Figured Bass Approach." In *Proceedings of the 7th International Conference on Digital Libraries for Musicology*, 27–31.
- Chapter 6: Ju, Yaolong, Sylvain Margot, Cory McKay, and Ichiro Fujinaga. 2020. "Figured Bass Encodings for Bach Chorales in Various Symbolic Formats: A Case Study." In *Music Encoding Conference Proceedings*, 71–73.

The candidate was responsible for all the experiments that were conducted in this research. The co-author Nathaniel Condit-Schultz was the principal writer of the manuscript for the second paper mentioned above (Condit-Schultz, Ju, and Fujinaga 2018), and implemented the rule-based algorithm introduced in Section 5.1. The remaining manuscripts for the papers, and the implementations of the algorithms introduced in this dissertation were all contributed by the candidate. The ground truth for the 39 Bach chorales introduced in Section 5.2.3.1 was prepared by the co-author Samuel Howes. The creation of the Bach Chorales Figured Bass (BCFB) dataset introduced in Section 6.2 was done by the candidate and co-authors Sylvain Margot and Luke Dahn collectively; the creation of the Bach Chorales Multiple Chord Labels (BCMCL) dataset in Section 6.5 was done by the candidate.

Table of Contents

List of Figures

List of Tables

Glossary

Ambiguity: A state of having more than one possible answer or interpretation (see Section 1.2).

Chord: Three or more unique pitch classes sounded simultaneously (or functioning as if sounded simultaneously) (see Section 1.3.2).

Chord labelling: The process of assigning chord labels for symbolic music. In this dissertation, it specifically means assigning chord letters (see Section 3.2.3).

Chord quality: The combination of the constituent intervals that form a chord (e.g., dominant 7^{th} chord, major triad, etc.).

Chord tone: A pitch class that makes up a chord.

Chord type: The identity of a chord which is jointly defined by chord root and chord quality. For example, if the twelve enharmonic pitch classes [C, C♯/D♭, D, D♯/E♭, E, F, F♯/G♭, G, G♯/A♭, A, A♯/B♭, B] are considered as chord root; the major and minor triads are considered as chord quality, there will be 24 different chord types, where the twelve different chord roots can be combined with the two kinds of chord quality.

Consonance and dissonance: The former refers to the sounding of sweetness, and the latter refers to the sounding of harshness. The distinction between them can be genre-, culture-, and era-dependent. In this dissertation, I consider vertical consonances and dissonances. Major third, minor third, major sixth, minor sixth, perfect fifth, perfect eighth, and unison are considered consonances, while other intervals (within an octave, since bigger intervals can be wrapped within an octave) are considered dissonances.

Figured bass: A type of music notation that uses numerals and other symbols to indicate intervals to be played above a bass note (see 3.2.1).

Harmony: Two or more unique pitch classes sounded simultaneously (or functioning as if sounded simultaneously) (see Section 1.3.2).

Homorhythm: Music with all parts sharing a very similar rhythm. It is commonly used in hymn and chorale settings.

Label distribution learning: A supervised machine learning paradigm where each sample of training data is associated with a distribution of membership scores for all label classes, and the resulting automated model is able to predict a distribution of membership scores for all label classes for each unseen sample (see Section 2.4).

Multi-label learning: A supervised machine learning paradigm where each sample of training data is associated with multiple labels, and the resulting automated model is able to predict multiple labels for each unseen sample (see Section 2.3).

Musical surface: The specific notes indicated on a musical score. Figured bass is not considered part of the surface.

Non-chord tone: A pitch class that does not make up a chord.

Note onset slice (slice): A way of describing the simultaneities of music, which is formed whenever a new note onset occurs in any musical voice, and each slice consists of the vertical set of notes sounding at that moment.

Pitch: A perceptual attribute that allows the ordering of sounds on a frequency-related scale extending from low to high.

Pitch class: The spelling of a pitch without octave information.

Single-label learning: A supervised machine learning paradigm where each sample of training data is associated with a single label, and the resulting automated model is able to predict a single label for each unseen sample (see Section 2.2).

Softmax: A function that normalizes the output vector of a machine learning classifier (often neural networks) as a probability distribution.

Sonority: The set of pitch classes present in a note onset slice.

Supervised machine learning: A process of learning from labelled data that contains input-output pairs, and predicts outputs based on new inputs. For example, it can be used in the task of machine English-French translation, where computers are trained based on English-French sentence pairs and can provide French translation for the new English sentences.

Symbolic music: Comprises any kind of score representation with an explicit encoding of notes or other musical events. These include machine-readable data formats such as MIDI, Music Encoding Initiative (MEI), and MusicXML.

Transitional motion: Notes that are either left or approached by step. In music theory, non-chord tones often exhibit such motions, including:

- Passing tone: Approached and departed by step in the same direction.
- Neighbour tone: Approached and departed by steps in opposite directions; the antecedent and consequent are the same note.
- Suspension/Retardation: Approached by unison (or sustain); departed by either down by a step (suspension) or up by a step (retardation); stronger metric position than antecedent.
- Appoggiatura: Approached by leap; departed by step in the opposite direction; stronger metrical position that antecedent.
- Escape tone: Approached by step; left by skip; weaker metrical position than its antecedent.
- Pedal tone: Approached by unison (or sustain); left by unison (or sustain).
- Double passing tone: Two notes of the same duration, separated by step; approached and departed by step in the same direction; the first of the pair must occupy a weaker beat than its antecedent.

Acronyms

BCFB: Bach Chorales Figured Bass

BCMCL: Bach Chorales Multiple Chord Labels

CNN: Convolutional Neural Networks

FBA: Figured Bass Annotation

LDL: Label Distribution Learning

LSTM: Long Short-Term Memory

MEI: Music Encoding Initiative

MIDI: Musical Instrument Digital Interface

MIR: Music Information Retrieval

ML: Machine Learning

MLL: Multi-Label Learning

NBA: Neue Bach Ausgabe

PC: Pitch Class

RB: Rule-Based

RNN: Recurrent Neural Networks

SLL: Single-Label Learning

SML: Supervised Machine Learning

Chapter 1 Introduction

1.1 Background and motivation

One of the focal points of this dissertation is chords, which are considered one of the most fundamental building blocks of Western tonal music. Throughout history, different theories of labelling chords have been proposed, and chord labels have been an important way of representing harmonic content. Chord label sequences have been foundational in not only identifying other high-level musical aspects, such as tonality and form, but also facilitating other music-theoretical and musicological studies, such as music genre analysis (Biamonte 2010) and characterizing the harmonic structure of a corpus (Moss et al. 2019). Obtaining chord labels at a large scale also serves various research purposes, including comparing harmonic patterns from different eras and facilitating harmonic queries in a searchable database. Despite being an essential analytical tool, manual chord labelling is a time-consuming process and requires years of training.

Automatic chord labelling presents a promising alternative. With the rapid development of computational power and the growing amount of annotated data, it has been possible to teach computers to conduct chord labelling automatically using supervised machine learning. One of the most typical supervised machine learning paradigms is to associate each example with a single label, known as single-label learning, which is often used in tasks with a single correct answer, such as audio-to-score alignment (Carabias-Orti et al. 2015), optical character recognition (Awel and Abidi 2019), and speech recognition (Prabhavalkar et al. 2017). However, in chord labelling, a chord does not always have a single label, and analyzing a musical passage can sometimes be ambiguous. Therefore, automatic chord labelling requires a different supervised machine learning paradigm than single-label learning to fully address this ambiguity.

The goal of this dissertation is to find out how tasks with ambiguity have been addressed in supervised machine learning and apply some of these methods to automatic chord labelling. To provide a broader context and background for this research, I will first discuss ambiguity in a general sense, which has been studied and discussed in other disciplines, such as visual art (Wimmer, Doherty, and Collins 2011), linguistics (Dayal 2004; Heim 2002), and music (Saslaw 1992; Agawu 1994).

1.2 Ambiguity: A brief overview

Ambiguity generally refers to the state of having more than one possible answer or interpretation. It can also be associated with *uncertainty*, indicating no definitive interpretation but multiple alternate ones. Ambiguity can be found in many fields. For example, *MB* in computer science means *megabyte*, which can be either decimal (10^6, 1,000,000 bytes) or binary (2^{20}, 1,048,576 bytes); the image shown in Figure 1-1 is a famous example of ambiguity in visual art, which can be seen as a young woman or an old woman.

Figure 1-1: My Wife and My Mother-In-Law, by the cartoonist W. E. Hill, 1915. Image from Nicholls, Churches, and Loetscher (2018).

There have been studies on the source of ambiguity (Mumpower and Stewart 1996; Aroyo and Welty 2014; Schaekermann et al. 2019), and one source of ambiguity is "under-specification", meaning the provided information is insufficient, leaving room for multiple possible interpretations and ambiguity (Schaekermann et al. 2019). For example, the task "find the cost of attending this university for undergraduates" is ambiguous, since there can be multiple answers depending on whether the student is in-state, out-of-state, or coming from a different country (Manam and Quinn 2018); even a simple math problem can be ambiguous: "List two positive

integers that sum up to four", the answer can be {1, 3} or {2, 2}, since it does not specify, for example, whether the two integers must be identical or not, leading to two possible answers.

Ambiguity also exists in natural languages and music, and they are often compared and considered to be related (Patel 2010). Next, I will overview the study of ambiguity in both fields as the steppingstone for addressing ambiguity in chord labelling, the main topic of this dissertation.

1.2.1 Ambiguity in natural languages

Ambiguity can be commonly found in natural languages. In fact, there have been extensive studies in linguistics on this matter, where different types of ambiguity, such as lexical ambiguity and syntactical ambiguity, are discussed (MacDonald 1993). *Lexical ambiguity* is the presence of two or more possible meanings for a single word. For example, the word "key" has more than one distinct definition, where the first one can be "a small piece of metal with incision cut" and the second one can be "a button on a computer keyboard, or a telephone", and the third one can be "solution or answer" as in "the key to happiness is being optimistic". *Syntactical ambiguity* means that there is more than one interpretation to a sentence because of its grammatical structure. For example, "He played the instrument on the floor", could mean that he played the instrument that was lying on the floor (as opposed to lying on the couch), or the sentence could mean that he was sitting on the floor while playing the instrument.

Both ambiguities can be understood as the result of "under-specification", as discussed in Section 1.2, since both ambiguities can be reduced or removed using more specifications, including context, other lexicons, or rephrasing to direct a concise and clear communication (MacDonald, Pearlmutter, and Seidenberg 1994).

1.2.2 Ambiguity in music

Ambiguity also exists in music. For example, the perceived music elements among subjects can be different. For the same piece of music, the pace of people's tapping to the music, for example, can be different, which is often in the form of half or double (Böck, Davies, and Knees 2019); this is also true in pitch perception, where certain harmonic complex tones can be ambiguous to identify which octave they belong (Normann, Purwins, and Obermayer 2001). Sherpard tone (Shepard 1964) is an example of this kind, and the succession of Sherpard tones can

result in pitch circularity, where a set of tones is perceived to have endless descending or ascending pitches (Deutsch, Dooley, and Henthorn 2008).

In symbolic music, ambiguity also exists in music analysis tasks, such as melody identification, key-finding, and cadence detection. These tasks have been a regular part of music theory and have been discussed by theorists over the centuries. One example is composite (compound) melody, which refers to a single melodic line that can be considered as a combination of several individual streams. A simple melodic line with two possible interpretations is shown in Figure 1-2.

Figure 1-2: Illustration of two interpretations of a melodic line, where the first one (left) contains two individual streams, and the second one (right) only contains one stream (Saslaw 1992, 117).

As one of the essential tasks of music analysis, chord labelling, also involves ambiguity, and I believe the reason, similar to natural languages in Section 1.2.1, is also "under-specification", meaning that the process of chord labelling has not been well defined, and ambiguity originates from different analytical perspectives in music theory. Analysts may adopt different analytical strategies, resulting in different chord labels. Next, I will introduce some basic music concepts, including *pitch*, *pitch class*, then *chord* and *harmony* in Section 1.3 that are essential to understand chord labelling, then a brief overview of the processing of chord labelling and its ambiguity will be introduced in Section 1.4.

1.3 Basic music concepts

1.3.1 Pitch and pitch class

Pitch is one of the most fundamental concepts in music, and as Klapuri (2006) defined: "*Pitch is a perceptual attribute which allows the ordering of sounds on a frequency-related scale*

extending from low to high." Basically, it is approximately proportional to log-frequency.[1] One way of representing pitch is MIDI number, and the formula that converts MIDI number (n) to its frequency (f_n) is:

$$n = 12 * log_2\left(\frac{f_n}{440}\right) + 69$$

One characteristic pitch has is *octave equivalence*, which is between pitches that are spelled the same but any number of octaves apart, and the spelling of a pitch without octave information is called *pitch class*. For example, the pitches C3 and C4 are considered as a pair of octave equivalence, where they are an octave apart and the pitch class of both pitches is C.

1.3.2 Chord

With the concept of *pitch class* in mind, *chord* is defined in this dissertation as follows:

Three or more unique pitch classes sounded simultaneously (or functioning as if sounded simultaneously).

This definition of *chord* applies specifically to Western tonal music from c. 1650 to c. 1900 known as the *common-practice period*, which is the scope of this study. Given the definition above, it is important to note that although a *chord* contains *three or more unique pitch classes*, not all *simultaneously sounding pitch classes* will comprise a *chord*, since some may be identified as *non-chord tones*, an essential step of chord labelling that will be introduced in Section 1.4.

Chords can often be understood as a construction from the intervals of (major and minor) thirds (Benward and Saker 2003), known as *tertian chords*.[2] Tertian chords with three unique pitch classes are *triads*, which are built by stacking two thirds on top of one another. Each third can either be major or minor interval, and the combination of two thirds result in four possibilities for triads:

- *major triad*: A minor third stacked on a major third;

[1] For details, a further extensive treatment of musical pitch can be found in Krumhansl (1990).
[2] Chords can also be constructed from the intervals of fourth and fifth intervals, known as *quartal* and *quintal* chords. Since they are generally rare in the common-practice period, the discussion of these chords is out of the scope of this dissertation.

- *minor triad*: A major third stacked on a minor third;
- *diminished triad*: A minor third stacked on a minor third;
- *augmented triad*: A major third stacked on a major third.

For example, a C major triad contains three unique pitch classes: C, E, and G. C is the starting pitch class on which the intervals are stacked upon and is known as the *chord root*; E and G are respectively considered as the *third*, and the *fifth* of the triad, since they form a third and a fifth interval above the chord root C. It contains a major third between C and E, and a minor third between E and G. Here, major, minor, diminished, and augmented are considered as the *chord qualities* of a triad, which is defined by the constituent intervals that form a chord.

Tertian chords with four unique pitch classes are *7th chords*, which are built by stacking a major or minor third on the fifth of a triad. Common 7th chord qualities in the common-practice period include:

- *major 7th chords*: A major third interval stacking on the fifth of a major triad;
- *minor 7th chords*: A minor third interval stacking on the fifth of a minor triad;
- *dominant 7th chords*: A minor third interval stacking on the fifth of a major triad;
- *diminished 7th chords*: A minor third interval stacking on the fifth of a diminished triad;
- *half-diminished 7th chords*: A major third interval stacking on the fifth of a diminished triad.

For example, a C major 7th chord contain four unique pitch classes: C, E, G, and B. B is considered as the *seventh* of the chord, since it forms a major seventh interval above the chord root C, and a major third interval above G, the fifth of the C major triad.

Other tertian chords with more unique pitch classes (e.g., 9th chords, 11th chords, etc.) can be found in the common-practice period, but they are generally rare and will not be discussed in detail here.

In the second part of the definition "functioning as if sounded simultaneously", it refers to the fact that there are pitch classes in a sequential order that can still be grouped and understood as a chord, since human listeners can perceptually integrate such a sequence into a single entity (Yeary 2011), in other words, a chord in this case. This is particularly evident in a musical passage

with arpeggiation, as the Alberti bass accompaniment shown in Figure 1-3, where notes of each measure are grouped and labelled as a chord.

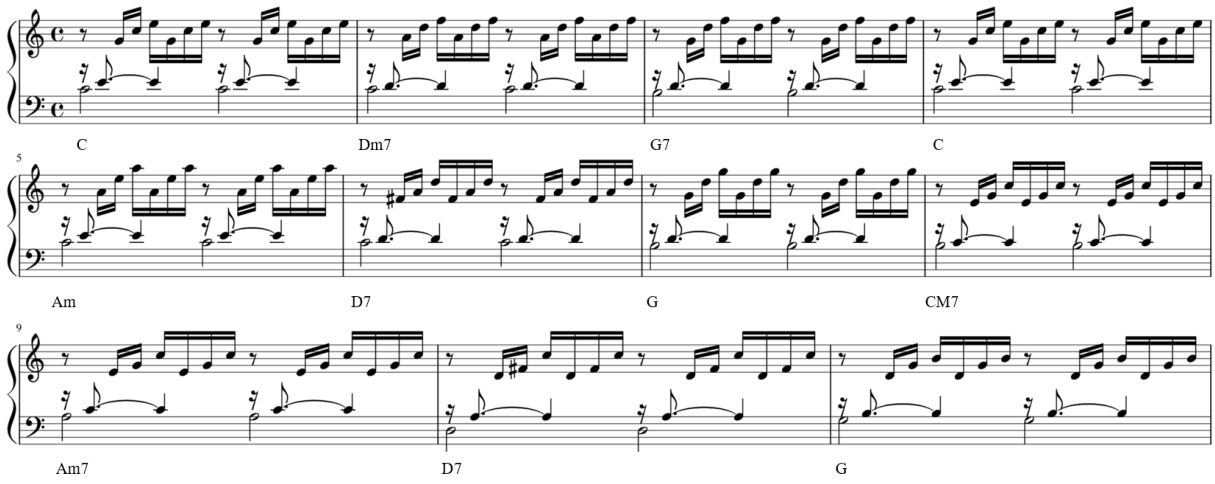


Figure 1-3: J. S. Bach Prelude in C, BWV 846, Measures 1–11 of the score with chord labels in the text below the lowest stave. Image modified from (Micchi, Gotham, and Giraud 2020).

Harmony can be understood as a more general case of chord, which can be defined as:

Two or more unique pitch classes sounded simultaneously (or functioning as if sounded simultaneously).

Harmony and *chord* can be considered synonyms in the common-practice period since harmony mostly exists in the form of three or more unique pitch classes. Before this period, harmony, however, could also exist in the form of two unique pitch classes, which can be mostly seen in the late medieval and early Renaissance eras. For details of harmony and its evolution through time, see Section 3.1.

1.4 Chord labelling and its ambiguity

As a way of representing harmonic content in music, chord labelling has been an essential part of music theory in Western tonal music. However, chord labelling can be ambiguous, and there is usually no best or only one correct way of chord labelling. During the common-practice period, different theories of chord labelling had been proposed, and experts may form their own ways of chord labelling, based on personal preferences, the musical training they received, and the

type of music to analyze. Next, I will introduce the process of chord labelling as three main steps and discuss how ambiguity can be involved in this process. These steps will be further elaborated in Sections 5.1.1.1, 5.1.1.2, and 5.1.1.3, respectively.

1. Decide what chord qualities should be considered, and the decision may vary depending on each individual analyst. Ambiguity will arise if two analysts have different chord qualities considered. For example, if one analyst considers 9th chords, and the other analyst only considers triads and 7th chords, their analyses will differ when ninths are involved in the music. We can resolve this ambiguity by proposing a definitive set of chord qualities for chord labelling, and in the common-practice period, for example, we can define that the chord qualities that should be considered are: Major, minor, diminished, and augmented triads; major, minor, dominant, half-diminished, and diminished 7th chords.

2. Identify non-chord tones. This step is to determine which notes in the musical surface do not comprise a chord. This step is the main source of chord labelling ambiguity, and the rules of whether a note is a chord tone or non-chord tone can be complex and inconsistent from one individual to another. For instance, not all simultaneities with a seventh interval will be labelled as 7th chords: Take the three chord labelling analyses of Figure 1-4 as an example, we can see that majority of the disagreements are due to the different interpretations of the sevenths: In m. 1.1.5,[3] for example, Analysis 2 labels "Em7" as the chord label, indicating that the pitch class "E" at Alto as a chord tone, while Analyses 1 and 3 suggest no chord change from the previous "G" chord, indicating "E" should be a non-chord tone (passing tone). The discussions of identifying non-chord tones are essential to this dissertation and will be further discussed in Sections 3.3.4, 4.1.2.1, and 5.1.1.2.

[3] m. 1.1.5 means the first measure, the first and half beat.

3. Mapping chord tones into chord labels. Once the choice of non-chord tones is made, the final step of chord labelling is to map the (remaining) chord tones into chord labels. Easy as it may seem, this step can sometimes involve ambiguity. Take the second beat, the first measure of Figure 1-4 as an example, we can see that although all three analyses consider all the notes of this beat as chord tones, disagreement in chord labels still arises: Analysis 2 labels m. 1.2 as a "D" chord since the seventh is not present until m. 1.2.5; while Analyses 1 and 3 label m. 1.2 as a "D7" chord, since the sonority of m. 1.2 is the subset of that of m. 1.2.5.

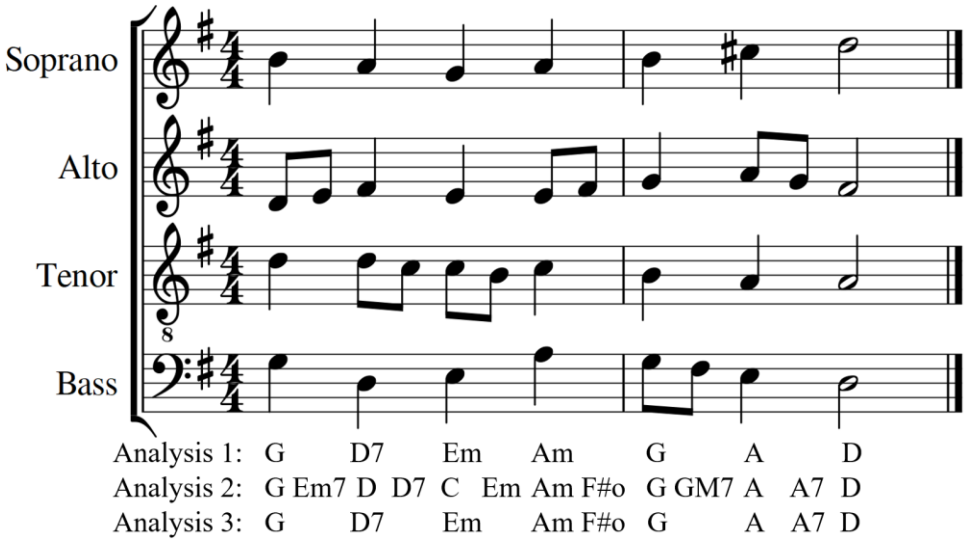


Figure 1-4: Illustration of a sample chorale passage that can be analyzed in three different ways. Image modified from Condit-Schultz, Ju, and Fujinaga (2018).

As shown above, ambiguity is involved in each of the three steps of chord labelling process, where analysts might adopt different analytical strategies, resulting in different chord label analyses. Note that Figure 1-4 only contains a small subset of all possible chord labelling analyses. It is used as a proof of concept showing how chord labelling ambiguity can result in multiple alternate chord labels. In this dissertation, I consider the fundamental cause for chord labelling ambiguity is under-specification, since the number of possible analyses can be greatly reduced if a clear analytical strategy for chord labelling (regarding each of the three chord labelling steps) is defined.

1.5 Previous work

Now that the chord labelling ambiguity has been discussed in a general sense, how has this ambiguity been addressed in automatic chord labelling? What approaches in supervised machine learning can be used to address this ambiguity in automatic chord labelling? This section will overview the potential options to answer these questions.

In supervised machine learning, there are three main approaches to address ambiguity (Schaekermann et al. 2020):

- The first approach views ambiguity as noise to be eliminated in the annotations (Schaekermann et al. 2020; Warby et al. 2014), so the results contain only a single answer for each sample. This can be achieved by, for example, using a more controlled group of annotators, carefully curating materials that are less ambiguous to analyze (Flexer and Lallai 2019), or using the majority vote among annotators where the most popular label is chosen (Voigt et al. 2010; Bien et al. 2018; Rajpurkar et al. 2018; Ruamviboonsuk et al. 2019; Sayres et al. 2019; Shibutani et al. 2019; Jukić, Kečo, and Kevrić 2018). Then, single-label learning is used as the supervised machine learning paradigm to build automatic models.
- Besides being a proxy of noise, ambiguity can also result from systematic preferences. Therefore, multiple answers can be aggregated for each sample to reflect the solution space of tasks that involve ambiguity (Zheng, Ma, and Huang 2018; Koops et al. 2020), and the resulting multiple parallel labels can be used to build automatic models using multi-label learning (Zhang and Zhou 2014).
- Label distribution learning: This is a more general case of multi-label learning, where each sample can be associated with not only multiple binary labels but also a distribution of membership scores for all possible classes, and each score represents the degree of fitness for that class (Geng 2016; Räbiger et al. 2018; Cohen et al. 2019).

The details of these approaches will be introduced in Chapter 2.

In the literature of chord labelling, there have been studies on how different chord labels provided by a group of human annotators can be (Ni et al. 2013; De Clercq and Temperley 2011; Koops et al. 2019). Although there are a few datasets with multiple chord label annotations (De

Clercq and Temperley 2011; Devaney et al. 2015; Koops et al. 2019) available, no efforts have been made to reduce these variabilities for automated models that employed single-label learning, and no attempts have been found that applied multi-label learning or label distribution learning to automatic chord labelling. A detailed literature review on automatic chord labelling will be given in Section 3.4.

1.6 Research questions and dissertation outline

In this dissertation, I will attempt to advance the study of automatic chord labelling by answering the following research questions:

- What is the fundamental cause of chord labelling ambiguity, and which part of the chord labelling process can result in ambiguity?
- If single-label learning is used for automatic chord labelling, what are the ways to generate single labels for chords?
- Is it possible to train automatic chord labellers to generate multiple labels simultaneously? If so, how?
- If there are multiple possible correct labels, are some still better/more fit than others? Can this be determined, quantified, and automated? If so, how?

In Section 1.4, I pointed out that the fundamental cause of chord labelling ambiguity is under-specification, and the number of possible analyses can be greatly reduced if a clear analytical strategy for chord labelling is defined. I also pointed out that there are three steps of chord labelling process that involve ambiguity: (1) Deciding the chord qualities, (2) identifying non-chord tones, and (3) mapping chord tones into chord labels. I will further support this idea by reviewing the chord labelling literature in Chapter 3, especially in Section 3.3, where I introduce chord labelling strategies from three prominent music theorists: Jean-Philippe Rameau (1683–1764), Gottfried Weber (1779–1839), and Heinrich Schenker (1868–1935), and, in particular, how they identify non-chord tones in chord labelling. In Chapter 4, I will discuss how chord labelling ambiguity has been discussed in both music theory (Section 4.1), and music information retrieval (Section 4.2).

In Chapter 2, I will introduce three supervised machine learning paradigms: Single-label learning, multi-label learning, and label distribution learning, the three approaches that have been

used to address ambiguity in supervised machine learning (introduced in Section 1.5). I will introduce the literature of automatic chord labelling research in Section 3.4 and will propose my methodologies of addressing ambiguity in automatic chord labelling in Section 4.3.

In Chapter 5, I will propose a rule-based algorithm that can generate chord labels using a specific analytical strategy as a way to produce single labels for chords in Section 5.1, In Section 5.2, these labels will be examined and modified by a music theory expert and will be used to build automatic chord labelling models using single-label learning.

In Chapter 6, I will obtain multiple parallel analyses from four new analytical strategies, where each strategy is essentially a rule-based algorithm that generates chord labels based on both figured bass annotations and the musical surface. In Chapter 7 (Section 7.2), these parallel analyses will facilitate multi-label learning of automatic chord labelling, so that the resulting models can generate multiple labels for each chord simultaneously. In Section 7.3, I will also explore label distribution learning for automatic chord labelling, so that the resulting models can predict not only multiple binary chord labels automatically, but also a distribution of membership/fitness scores for all chord label classes. Finally, all the research involved in this dissertation and all the answers to the research questions will be summarized in Chapter 8, with possible directions for future research.

Chapter 2 Addressing ambiguity in supervised machine learning: A literature review

2.1 Introduction

With the rapid development of computational power and the growing amount of annotated data, supervised machine learning approaches have become popular in the field of information retrieval. One typical paradigm is to associate each example with a single label, known as single-label learning, which is often used in tasks with a single correct answer, including audio-to-score alignment (Carabias-Orti et al. 2015), optical character recognition (Awel and Abidi 2019), and speech recognition (Battenberg et al. 2017).

Many tasks in real life, however, do not always have a single correct answer: A song might belong to multiple musical genres, a sentence can be translated in numerous ways from one language to another, a facial expression may exude multiple emotions, etc. In these cases, ambiguity occurs when an example can be assigned more than one label (class). These ambiguous cases can potentially be associated with multiple binary labels, or possibly a distribution of membership scores for all the labels. The former is known as multi-label learning (MLL, Zhang and Zhou 2014) and the latter is known as label distribution learning (LDL, Geng 2016).

In Section 1.5, I listed three main approaches that use single-label learning, multi-label learning, and label distribution learning, to address ambiguity in supervised machine learning. Here, I will expand on each of these approaches, in Sections 2.2, 2.3, and 2.4, respectively. I will focus particularly on each approach's methodology and how it has been attempted before. Additionally, there is another supervised machine learning paradigm called *graded multi-label learning* that is situated between MLL and LDL, and which often has equal or more than three possible values for each class. These values can be understood as *graded memberships,* since each class can have different levels of membership (e.g., "not at all", "somewhat", "almost", and "fully") (Cheng, Hüllermeier, and Dembczynski 2010). Graded multi-label learning is introduced in Section 2.5.

2.2 Single-label learning

In single-label learning (SLL), tasks are assumed to have one definitive answer. Many of these tasks are unambiguous, where the correct answer is available (e.g., audio-to-score alignment using synthesized audio generated from the score) or can be easily inferred with no disagreement (e.g., optical character recognition). However, sometimes neither is the case, and this is especially true in medical fields, such as assessing bone age for unknown samples (Dallora et al. 2019), detecting tumour boundaries, (Zhao and Xie 2013), and pathology diagnosis (Rajpurkar et al. 2018), or perhaps in music information retrieval, where the true composer of a piece is unknown. In these situations, ambiguity arises when experts' estimates of the correct answer are non-unanimous. In order to collect ground truth data, ambiguity is often addressed by choosing the most common estimate as the approximation of the "correct answer"; these tasks are then treated as an SLL problem, which assigns only one label to each example. This approach can often be found in the medical domain, including diagnosing abnormalities for vocal fold paresis (Voigt et al. 2010), diabetic retinopathy (Sayres et al. 2019), and knee injuries (Bien et al. 2018).

2.3 Multi-label learning

As discussed in Section 2.1, multi-label learning (MLL) paradigm can be used for tasks that inherently lack a single definitive answer, but rather have multiple acceptable labels for each example. For instance:

- In machine translation, a sentence can have multiple valid translations when translated from one language to another.
- In text classification, an article can cover multiple topics simultaneously, such as media, finance, and politics.
- In genre classification, a song can belong to multiple musical genres.
- In key finding, the musical passage that prepares the modulation can have two possible answers of key.

In these tasks, ambiguity can be addressed by incorporating multiple acceptable answers, so that each example can be associated with multiple parallel annotations when applicable. One of

the early applications of MLL is text categorization (McCallum 1999; Y. Yang and Liu 1999; Schapire and Singer 2000). Over time, MLL has been receiving an increasing amount of attention, and many other tasks began to adopt this paradigm, such as music genre classification (Sanden and Zhang 2011), music emotion recognition (Trohidis et al. 2008), other multimedia contents (Qi et al. 2007; Fang Zhao et al. 2015), and tagging/tag recommendation (Katakis, Tsoumakas, and Vlahavas 2008; Jain, Prabhu, and Varma 2016; Shrivaslava et al. 2020). Next, I will introduce the methodologies for multi-label learning.

2.3.1 Formal definitions of multi-label learning

To introduce the methodology of MLL, I use the following definitions: Consider D as a dataset with N examples. Each example $E_i = (X_i, Y_i), i = 1, \dots, N$ is associated with a feature vector $X_i = (x_{i1}, x_{i2}, \dots, x_{iM})$, which contains M features, and a set of labels $Y_i \in L$, where $L = \{y_1, y_2, \dots, y_Q\}$ contains all Q possible labels. The goal of MLL is to learn a mapping $h(X_i) \rightarrow Y_i$, which is able to predict a set of labels Y_i based on an unseen example X_i.

2.3.2 Main challenge

Traditional SLL can be considered a special case of MLL, where each example is always associated with a single label, and MLL can also associate each example with multiple labels (when applicable). The greater generality of MLL also leads to its main challenge: Multi-label annotations can lead to an overwhelming size of output space, where the number of possible label combinations grows exponentially as the number of candidate classes increases. For example, an output space with 40 candidate classes ($Q = 40$) will result in more than a trillion possible label combinations ($2^{40} = 1{,}099{,}511{,}627{,}776$).

To mitigate this problem, one possible solution is to exploit *correlations* among labels, since not all label combinations are equally likely to occur. For example, the probability of a song with the label *pop* will be high if it is already tagged with *Michael Jackson* or *90s*; a paragraph will be unlikely to be labelled as *scientific* if it is related to *entertainment*. Therefore, it is beneficial to exploit label correlations for MLL problems. Overall, existing techniques can be categorized into three groups, based on the *order of correlations* considered (Zhang and Zhang 2010; Zhang and Zhou 2014):

- *First-order strategy*: MLL can be either addressed using algorithms that directly process multi-label data, such as multi-label *k*-nearest neighbour (Zhang and Zhou 2007) or multi-label decision tree (Clare and King 2001) classifiers, or simplified as a combination of independent binary classifiers (one per class, also known as "binary relevance") (Boutell et al. 2004; Clare and King 2001; Zhang and Zhou 2007). The main advantage of this approach is its simplicity and high efficiency. As a downside, it completely ignores label correlations.
- *Second-order strategy*: MLL can also be addressed in a pair-wise fashion, such as considering any possible pair of labels (Ghamrawi and McCallum 2005; Qi et al. 2007; Zhu et al. 2005), or ranking all the labels using pair-wise comparison (Fürnkranz et al. 2008). These second-order strategies consider label correlations to some degree, which can result in better performance. However, real-world applications can be complex, and label correlations can easily exceed the second-order assumption, where one label is assumed to be only related to another label (Zhang and Zhou 2014).
- *High-order strategy*: There are a few ways to model high-order label correlations in MLL, including considering all other labels' influences on each label (Cheng and Hüllermeier 2009; Godbole and Sarawagi 2004; Ji et al. 2008; Yan, Tesic, and Smith 2007) or considering connections among different subsets of labels (Read, Pfahringer, and Holmes 2008; Read et al. 2011; Tsoumakas and Vlahavas 2007). The advantage is that the high-order strategy considers more label correlations than the first-order and second-order strategies, but is computationally more expensive.

To implement these strategies, I can either transform MLL into a traditional SLL problem or develop specialized algorithms that can process multi-label data directly. These two approaches will be introduced in Section 2.3.3 and Section 2.3.4, respectively.

2.3.3 Problem transformation

In this approach, MLL problems are transformed into traditional SLL problems, where single-label classification algorithms can be used. Depending on the orders of label correlations considered, a variety of machine learning techniques can be used.

2.3.3.1 First-order strategy

If no label correlations are considered, MLL can be modelled as a combination of binary classifiers (known as binary relevance), where each class is predicted using a binary classifier. The main advantage of this approach is its high efficiency and simplicity in handling multi-label data, which only requires Q independent binary classifiers and has been used in different MLL applications (Boutell et al. 2004; Clare and King 2001; Zhang and Zhou 2007).

2.3.3.2 Second-order strategy

MLL can also be addressed in a pair-wise fashion by using chains of binary classifiers, where the subsequent one is dependent on the previous one's predictions (Read et al. 2011). For an unseen example, the predicted set of labels is obtained by traversing the chains of classifiers iteratively. This strategy considers the direct, second-order correlation between the current label and the previous label.[4] However, errors can propagate through this chain-like implementation.

2.3.3.3 High-order strategy

High-order correlations among labels are typically approached using MLL in two ways. The first way is called *label powerset (LP)*, which transforms MLL into SLL in three steps (Tsoumakas and Vlahavas 2007; Tsoumakas, Katakis, and Vlahavas 2010):

- First, LP enumerates all the existing combinations of labels in the training data and assigns each to a single class, which transforms the multi-label data into a single-label representation.
- Then, traditional SLL classification algorithms are used to predict a single label for the unseen example.
- Finally, the predicted single label is then transformed back to the multi-label representation using the inverse of the same mapping from the first step.

[4] Some also believed that this approach implicitly considered the high-order label correlations, since the current predicted label relies on all the previous predictions iteratively, as suggested by Zhang and Zhou (2014).

Despite its simplicity, LP has three drawbacks (Zhang and Zhou 2014). LP can only predict the label sets found in the training data, which may imperfectly generalize to combinations found in the unseen test data. LP can also be highly inefficient: When there are numerous label combinations, the resulting number of classes can be overwhelmingly big, leading to high complexity in training SLL algorithms. Furthermore, each class may have fewer training examples (Tsoumakas and Vlahavas 2007), which might lead to the problem of data scarcity.

The second way of modelling high-order label correlations is called *Random k-Labelsets*. As a modification of LP, Random k-Labelsets uses ensemble learning (Dietterich 2000; Zhou 2012) that involves a group of LP classifiers. The key is to apply each LP classifier only to random *k-labelsets* (size-*k* subsets of all possible label combinations) to improve computational efficiency, and to combine a group of LP classifiers to cover all the possible label combinations (Zhang and Zhou 2014).

2.3.4 Algorithm adaptation

Here, existing SLL algorithms are modified so that they can process multi-label data directly, such as multi-label decision trees (Clare and King 2001), rank support vector machines (Elisseeff and Weston 2002), collective multi-label classifiers (Ghamrawi and McCallum 2005), or neural networks (Zhang and Zhou 2006; Nam et al. 2014). Most of these methods, however, either capture only the first-order or second-order label correlations (Zhang and Zhou 2014).

In recent years, recurrent neural networks (RNN) have been used in MLL to model high-order label correlations. For example, Chen et al. (2017) combined convolution neural networks (CNN) with RNN for text categorization, and Yang et al. (2018) modified the attention framework from Vaswani et al. (2017), using an adjusted long-short term memory (LSTM) decoder for MLL (abbreviated as MLL-attention). In particular, Yang et al. (2018) compared MLL-attention against five baseline MLL algorithms: Linear SVM used in binary relevance (Boutell et al. 2004), classifier chains (Read et al. 2011), label power set (Tsoumakas and Katakis 2007), CNN used in binary relevance (Kim 2014), and CNN-RNN proposed by Chen et al. (2017). Results showed that their MLL-attention model achieved better performance than the baseline models in text categorization and subject identification.

2.3.5 Evaluation metrics

Section 2.3.3 and Section 2.3.4 introduced different MLL algorithms that predict multiple labels for each example. For evaluation, both example-based and class-based metrics can be used, as explained below.

2.3.5.1 Example-based metrics

In example-based metrics, some measure of performance is calculated for each test example, and the mean value across the whole test set is considered as the overall performance of an MLL model on the given dataset. Considering P as the size of the test set, five example-based metrics will be introduced (Ghamrawi and McCallum 2005; Godbole and Sarawagi 2004; Schapire and Singer 2000; Gibaja and Ventura 2015). As an illustration, assume there are only two test examples, where the predicted labels are [0, 1, 1, 0, 0] and the ground truth labels are [0, 1, 0, 1, 1] for the first example; and the predicted labels are [0, 1, 1, 0, 0] and the ground truth labels are [0, 1, 1, 0, 0] for the second example. The following equations adopt the notations introduced in Section 2.3.1:

- $Subset\ accuracy = \frac{1}{P}\sum_{i=1}^{P}[h(X_i) = Y_i]$,[5] which indicates the fraction of correctly predicted test examples. The prediction is correct only if the label set is identical to that of the ground truth. In this case, $Subset\ accuracy = \frac{1}{2}(0 + 1) = 0.5\ (50\%)$ since only the prediction of the second test example is correct.
- $Hamming\ loss = \frac{1}{P}\sum_{i=1}^{P}\frac{1}{Q}h(X_i) \triangle Y_i$, where $\triangle$ indicates the number of predicted classes that are different from the ground truth. In this case, $Hamming\ loss = \frac{1}{2} \times \frac{1}{5}(3 + 0) = 0.3\ (30\%)$, since there are three different classes in the first example and zero in the second example.

[5] Here, the expression within the bracket will result in a value of 1, if it is true, and 0 otherwise.

- $Precision = \frac{1}{P}\sum_{i=1}^{P}\frac{|Y_i \cap h(X_i)|}{|h(X_i)|}$, which shows the fraction of how many positive predictions (those with 1s) are correct. In this case, $Precision = \frac{1}{2} \times \left(\frac{1}{2} + \frac{2}{2}\right) = 0.75$ (75%), since the algorithm made two positive predictions (one correct) for the first example, and two (both correct) for the second example.
- $Recall = \frac{1}{P}\sum_{i=1}^{P}\frac{|Y_i \cap h(X_i)|}{|Y_i|}$, which shows the fraction of how many of the positive labels in the ground truth are covered. $Reall = \frac{1}{2} \times \left(\frac{1}{3} + \frac{2}{2}\right) = \frac{2}{3} \approx 0.667$ (66.7%), since the algorithm made one correct prediction (out of three in the ground truth) for the first example, and two out of two correct in the second example.
- $F1 = \frac{2*precision*recall}{precision+reeall}$. In this example, the $F1 = \frac{2*0.75*\frac{2}{3}}{0.75+\frac{2}{3}} = \frac{12}{17} \approx 0.706$ (70.6%).

2.3.5.2 Class-based metrics

In class-based metrics, the performance for each class over all the test examples *P* is calculated, and the MLL model's performance is the mean value of the performances across all the classes, *Q*. For the *j*-th class y_j, four basic metrics quantifying the binary classification performance on each label can be defined (Zhang and Zhou 2014):

$$True\ positive\ (TP_j) = \left|\{X_i | y_j \in Y_i \wedge y_j \in h(X_i), 1 \le i \le N,\ 1 \le j \le Q\}\right|;$$

$$False\ positive\ (FP_j) = \left|\{X_i | y_j \notin Y_i \wedge y_j \in h(X_i), 1 \le i \le N,\ 1 \le j \le Q\}\right|;$$

$$True\ negative\ (TN_j) = \left|\{X_i | y_j \notin Y_i \wedge y_j \notin h(X_i), 1 \le i \le N,\ 1 \le j \le Q\}\right|;$$

$$False\ negative\ (FN_j) = \left|\{X_i | y_j \in Y_i \wedge y_j \notin h(X_i), 1 \le i \le N,\ 1 \le j \le Q\}\right|,$$

where:

$$TP_j + FP_j + TN_j + FN_j = P.$$

Then, precision, recall, and F1-score can be calculated for each class y_j individually:

$$Precision_j = \frac{TP_j}{TP_j + FP_j}$$

$$Recall_j = \frac{TP_j}{TP_j + FN_j}$$

$$F1_j = \frac{2 * Precision_j * Recall_j}{Precision_j + Recall_j}$$

The precision, recall, and F1 can be calculated in two ways (Zhang and Zhou 2014), either by averaging all the $Precision_j$, $Recall_j$, and $F1_j$ values, which is known as *macro-averaging*:

$$MacroPrecision = \frac{1}{Q}\sum_{j=1}^{Q} Precision_j$$

$$MacroRecall = \frac{1}{Q}\sum_{j=1}^{Q} Recall_j$$

$$MacroF1 = \frac{1}{Q}\sum_{j=1}^{Q} F1_j$$

or by summing up all the $TP, FP, TN,$ and FN values across all the classes, which is knows as *micro-averaging*:

$$MicroPrecision = \frac{\sum_{j=1}^{Q} TP_j}{\sum_{j=1}^{Q} TP_j + \sum_{j=1}^{Q} FP_j}$$

$$MicroRecall = \frac{\sum_{j=1}^{Q} TP_j}{\sum_{j=1}^{Q} TP_j + \sum_{j=1}^{Q} FN_j}$$

$$MicroF1 = \frac{2 * MicroPrecision * MicroRecall}{MicroPrecision + MicroRecall}$$

We can see that *macro-averaging* assigns each class equal weight, while *micro-averaging* assigns more weight to the more frequent classes. They can both be used in MLL to reflect different aspects of a model's performance.

2.4 Label distribution learning

MLL assigns multiple binary labels to an example. However, many real-world tasks do not fit in with this binary positive-negative paradigm and demand a continuous value between 0 and 1 to shows how much each label describes the example. According to Geng (2016), the biological experiments on the genes of yeast over time may display various gene expression levels (Eisen et al. 1998); emotions analyzed from facial expression are often a combination of multiple basic emotions, such as happiness, sadness, surprise, anger, and fear, where the intensities of all the basic emotions can form a distribution of emotions for facial expression (Yin et al. 2006). In these tasks, a natural way to label an example X_i is to assign a real number $d_{y_j} \in [0, 1]$ for each class y_j, representing the degree to which y_j corresponds to X_i as membership scores (Geng 2016).[6] The membership scores for all classes will add up to 1, and this framework is known as label distribution learning (LDL), a generalization of MLL and SLL where ambiguity cannot be sufficiently addressed by assigning one or multiple binary values to each label (Geng 2016).

Geng (2016) notes that SLL and MLL can be used to answer the question "*which label(s) can describe the instance?*", but that neither of them can answer the further question "*how much does each label describe the instance?*" Then, he presented the framework of LDL and a comparison with SLL and MLL is shown in Figure 2-1, which mainly differ in three aspects (Geng 2016):

- LDL explicitly associates each training example with a label distribution, not a single label (SLL) or a label set (MLL).

- During the process of prediction, most SLL and MLL algorithms generate a numerical value for each class, which is later used to output the positive label(s) using a threshold. The specific value of each class does not matter as long as all the positive label(s) passes the threshold, while the numerical value in LDL is essential to predicting a label distribution.

[6] The membership score is best to represent as $d_{X_i}^{y_j}$, but here I use d_{y_j} for the sake of simplicity.

- The evaluation metrics for SLL and MLL algorithms (e.g., accuracy, precision, recall, and F1-score) are significantly different from those for LDL, which are often based on similarity or distances between the predicted distributions and the ground truth (real) distributions. These LDL metrics will be further discussed in Section 2.4.2.

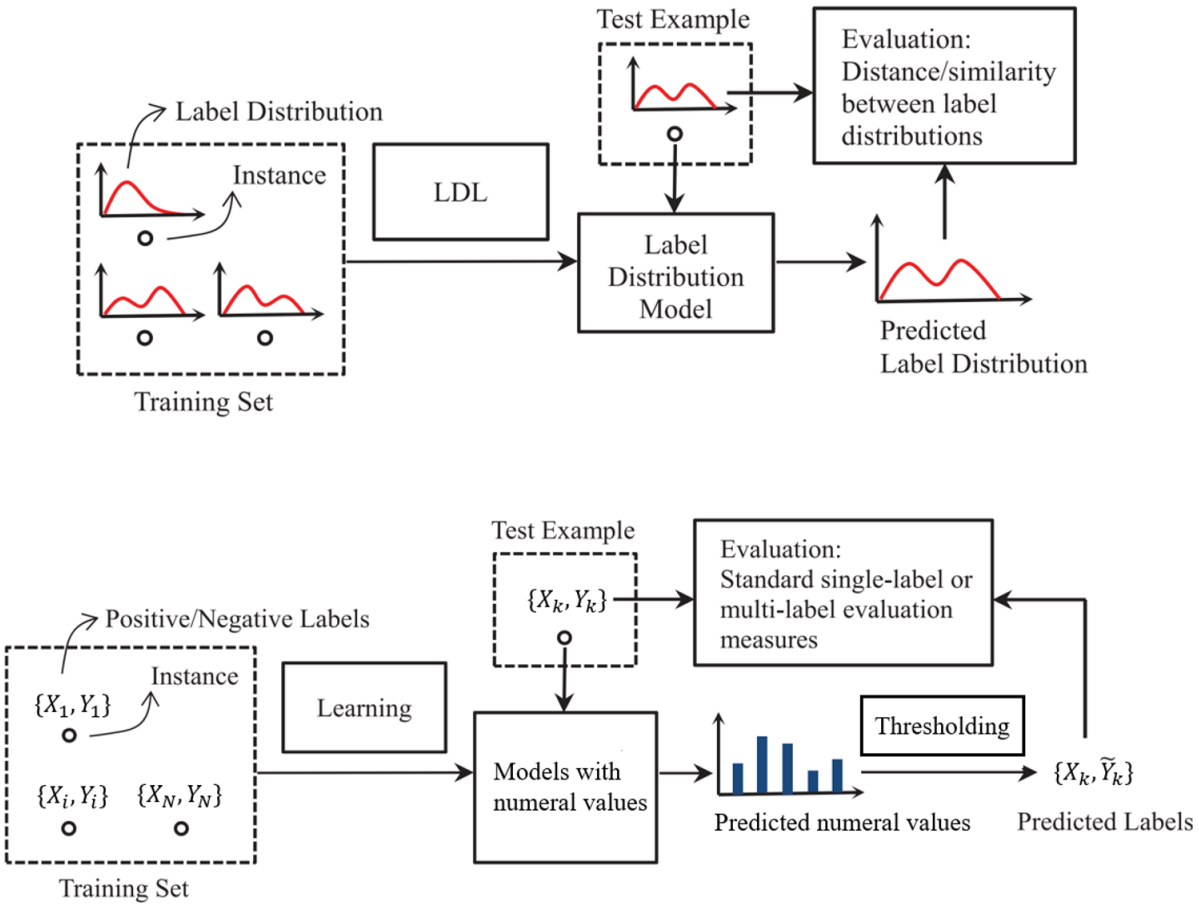


Figure 2-1: Diagrams of the training and testing of LDL algorithms (top) as well as SLL and MLL algorithms (bottom), based on Geng's diagrams (2016).

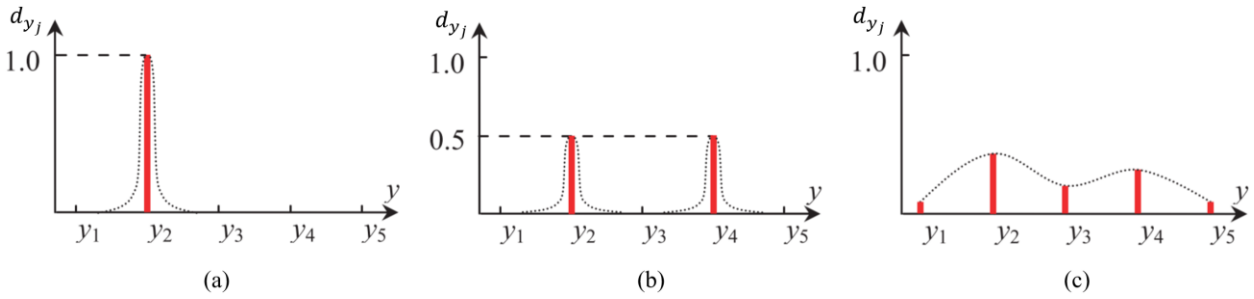


Figure 2-2: Label distribution examples for annotations of SLL (a), MLL (b), and LDL (c), based on Geng's diagrams (2016). The horizontal axis lists all classes, and the vertical axis is the probability (or relevance) of that class.

The annotations of SLL, MLL, and LDL are shown in Figure 2-2. For SLL, only the class y_2 is positive ($d_{y_2} = 1$), meaning that the label y_2 fully describes the example X_i; for MLL, two classes y_2 and y_4 are both positive ($d_{y_2} = d_{y_4} = 0.5$), meaning y_2 and y_4 both describe the example X_i equally; the LDL annotation offer more flexibilities of how labels are distributed, as long as d_{y_j} is a real number $\in [0, 1]$, and $\sum_{j=1}^{Q} d_{y_j} = 1$. This example demonstrates that SLL and MLL can be seen as special cases of LDL, and the general case of LDL offers more versatility in processing real-world data and annotations (Geng 2016).

2.4.1 Learning algorithms

Similar to MLL, LDL can either be transformed into the traditional SLL problem, using approaches such as Bayes classifier (Geng 2016) or support vector machines (SVM) (Wu, Lin, and Weng 2004; Lin, Lin, and Weng 2007), or one can adapt existing learning algorithms to process label distribution data directly, such as k-nearest neighbour (*k*NN) classifiers (Geng 2016) or neural networks (NN). The goal of these algorithms is to predict a membership score (denoted as $\tilde{d}_{y_j}$) as close to the ground truth (denoted as d_{y_j}) as possible. As an example, let us take the case of an adapted NN (Gao et al. 2017):

- After calculating the activation values of the NN's output layer, LDL applies softmax (see Glossary) as normalization to the output layer, and the resulting normalized distribution of numerical values will be considered as the final output, while SLL and MLL will apply a threshold to obtain the positive label(s).

- Since the aim of LDL is to approximate a label distribution rather than binary, positive labels, a different loss function is therefore required. For example, Kullback-Leibler (KL) divergence $KL\left(Y_i, \tilde{Y}_i\right) = \sum_{j=1}^{Q} d_{y_j} \ln \frac{d_{y_j}}{\tilde{d}_{y_j}}$ can be used to measure the difference between the predicted (denoted as tilde) and the ground truth label distributions (Geng 2016; Gao et al. 2017); in this case the best parameter set θ^* of NN is chosen by minimizing the following difference with the ground truth (Gao et al. 2017):

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_j d_{y_j} \ln \frac{d_{y_j}}{\tilde{d}_{y_j}} = \underset{\theta}{\operatorname{argmin}} - \sum_j d_{y_j} \ln \tilde{d}_{y_j}.$$

  Therefore, the loss function can be defined as:

$$L = - \sum_j d_{y_j} \ln \tilde{d}_{y_j},$$

  from which the derivatives can be inferred for the backpropagation algorithm to optimize all the NN's parameters.

There are also specialized algorithms explicitly developed for LDL, such as SA-IIS (Specialized Algorithm-Improved Iterative Scaling) and SA-BFGS (Specialized Algorithm-Broyden–Fletcher–Goldfarb–Shanno) (Geng 2016).

2.4.2 Evaluation metrics

To measure the difference between the predicted and the ground truth label distributions, there are two main groups of evaluation metrics available (Geng 2016):

- Distance metrics: Higher distance indicates a greater difference between the two distributions. Four distance metrics are introduced below:
  - Chebyshev distance (Cheb) examines all the classes and picks out the one whose predicted value is the most different from the ground truth.

$$Cheb\left(Y_i, \tilde{Y}_i\right) = \max_j \left| d_{y_j} - \tilde{d}_{y_j} \right|$$

- Clark distance (Clark) and Canberra metric (Canber) are more sophisticated metrics, where the difference for each class is considered:

$$Clark\left(Y_i, \tilde{Y}_i\right) = \sqrt{\sum_{j=1}^{Q} \frac{(d_{y_j} - \tilde{d}_{y_j})^2}{(d_{y_j} + \tilde{d}_{y_j})^2}}$$

$$Canber\left(Y_i, \tilde{Y}_i\right) = \sum_{j=1}^{Q} \frac{\left|d_{y_j} - \tilde{d}_{y_j}\right|}{d_{y_j} + \tilde{d}_{y_j}}$$

- Kullback-Leibler divergence (KL): As discussed above, this can serve during the training of neural networks in calculating the loss function in LDL. KL can also be used as an evaluation metric:

$$KL\left(Y_i, \tilde{Y}_i\right) = \sum_{j=1}^{Q} d_{y_j} \ln \frac{d_{y_j}}{\tilde{d}_{y_j}}$$

- Similarity metrics: Higher similarity indicates a smaller difference between the two distributions. Two similarity metrics are introduced below:

  - Cosine coefficient (Cosine):

$$Cosine\left(Y_i, \tilde{Y}_i\right) = \frac{\sum_{j=1}^{Q} d_{y_j} \tilde{d}_{y_j}}{\sqrt{\sum_{j=1}^{Q} d_{y_j}^2} \sqrt{\sum_{j=1}^{Q} \tilde{d}_{y_j}^2}}$$

  - Intersection similarity (Intersec), which is the sum of all the smaller values between y_j and $\tilde{y}_j$:

$$Intersec\left(Y_i, \tilde{Y}_i\right) = \sum_{j=1}^{Q} \min\left(d_{y_j}, \tilde{d}_{y_j}\right)$$

2.5 Graded multi-label learning

As mentioned in Section 2.1, *graded multi-label learning* is situated between MLL (each class has a binary positive/negative value) and LDL (each class has a real value between 0 and 1),

since graded multi-label learning has equal or more than three possible membership values (e.g., "not at all", "somewhat", "almost", and "fully") (Cheng, Hüllermeier, and Dembczynski 2010) for each class (Brinker, Mencía, and Fürnkranz 2014).[7]

Similar to MLL, GMLL can either use a multi-class classifier for each class (first-order strategy discussed in Section 2.3.2) or cascade these classifiers as chains (second or high-order strategies) to transform into a traditional SLL problem (Cheng, Hüllermeier, and Dembczynski 2010). GMLL also has specialized algorithms developed specifically for it. One example is shown by Cheng, Hüllermeier, and Dembczynski (2010), who modified the existing MLL algorithm IBLR-ML (Instance-Based Learning by Logistic Regression-Multi-Label) to process graded multi-label data directly.

2.6 Conclusion

In this chapter, I introduced four paradigms that can be used to address ambiguity in supervised machine learning: Single-label learning (SLL), multi-label learning (MLL), label distribution learning (LDL), and graded multi-label learning (GMLL). SLL usually attempts to address ambiguity by choosing a single label as ground truth during the process of data collection. MLL, LDL, and GMLL usually embrace ambiguity by incorporating experts' annotations, as multiple binary values, label distributions, or multiple graded values. All four approaches can be approached using specialized algorithms, and the three last approaches can be reformulated to be treated as if they were an SLL problem.

Next, I will introduce a literature review of chord labelling in Chapter 3. In Chapter 4, I will combine the nuance of this chapter and Chapter 3, and discuss possible ways of addressing ambiguity in automatic chord labelling. Then, SLL will be used in Chapter 5, and MLL as well as LDL will be used in Chapter 7. I consider GMLL a fine-grained approach between MLL and LDL for automatic chord labelling and have not experimented with it in this dissertation.

[7] When there are only two possible membership values (as an exception), it reduces to a multi-label learning problem (Brinker, Mencía, and Fürnkranz 2014).

Chapter 3 Literature review of chord labelling

The practice of chord labelling can be traced back to the late sixteenth and early seventeenth century, when composers began to write numerals and other symbols above the bass to indicate chords. During the same period, chords were recognized as the basic building blocks of harmony and remained an indispensable part of Western tonal music throughout the common practice period (1650–1900). In this chapter, a brief history of harmony will be introduced in Section 3.1, and in Section 3.2 I will present an overview of different chord label representations.

As a key part of Western music theory, chord labelling is an essential step in music analysis, which can be used to discover a composer's style and identify other musical features, such as tonality and form. Nowadays, it is commonly taught in music schools and has become a required skill for composers, musicians, and theorists. There are different ways to label chords, and many prominent music theorists (e.g., Rameau, Weber, and Schenker) have proposed approaches of their own, and some of these approaches will be introduced in Section 3.3.

Chord labelling is a complex process, which requires years of training and makes the acquisition of manually annotated chord labels extremely expensive. Automated models using heuristics and supervised machine learning present a promising alternative, and an associated literature review of automatic chord labelling will be introduced in Section 3.4.

3.1 A brief history of harmony

The history of harmony is fairly long and complex. Since this dissertation is mostly concerned with harmony in the common practice period (from c. 1600 to c. 1900), I will introduce the history of harmony before this period (from Ancient Greece to Renaissance, see Section 3.1.1) briefly, then expand more for the common practice period (Section 3.1.2).

3.1.1 Harmony before the common practice period

In history, the definition of "harmony" has changed over time. In Greek music, harmony referred to the succession of notes within an octave, and in the Middle Ages, it referred to the simultaneous sounding of two notes (Cohn et al. 2001). One of the earliest documented examples

of harmony (simultaneous sounding of two or more notes) in the history of Western Europe is found in the last half of the ninth century, when the choirs performing in many churches began to add an extra voice that moved in exact parallel[8] with the original melody at an interval of perfect fourth or fifth, known as *organum* (Fuller 2002, 480). Through time, polyphony gained more flexibilities and developed into other forms, which can contain multiple additional voices moving in opposite or contrary motion to the given melody, at a wider variety of intervals, and at different rhythmic values (Fuller 2002, 485–490). *Counterpoint*, for example, is a form of polyphony developed in the fourteenth century (Fuller 2002, 490).

At the end of the sixteenth century, there was a change in musical style: Although the basic style of composition was primarily concerned with counterpoint which largely regards note-against-note texture and intervals (Fuller 2002, 477), chords that were formed from the stackings of intervals had become the basic harmonic units. This resulted in a shift of the conception of harmony, where theorists such as Gioseffo Zarlino (1517–90) considered chords as the fundamental building blocks of harmony (Lester 1994, 754).

3.1.2 Harmony in the common practice period (c. 1600–c. 1900)

3.1.2.1 Harmony as figured bass: The Baroque Era (c. 1600–c. 1750)

As chords became the basic unit of harmony at the end of the Renaissance period and the beginning of the common practice period (around 1600s), the notation of figured bass appeared (Lester 1994, 49), which used numerals and other symbols written above the bass line to indicate accompanying chords to be played. It is a combination of the bass line and figures, also known as *thorough bass*, that enable accompanying instrumentalists to improvise or realize a complete harmonic support for the melody (often the top-most voice). The realized accompaniment (including the bass line) is called *basso continuo*.

The concept of chords in this era is significantly different from the one we understand today. In the figured bass notation system, the identity of a chord is defined by figures (Bach [1753] 1949).[9] For example, chords with 5/3, 6/3, or 6/4 figures are considered as different triads.

[8] This indicates that the added voice shares the same rhythm with the original melody.
[9] In this dissertation, I will use *slash* (/) to indicate chords with multiple figures.

Although they can be the same chord in different inversions[10], this concept was not formalized until Rameau's treatise on harmony published in 1722 (Lester 2002, 759). Furthermore, chord qualities (i.e., major or minor third) are not explicitly indicated and have to be inferred based on the bass and key.

Despite this difference, figured bass still offers some indications of harmonic rhythm, serving as a promising basis to approach chord labelling (see Section 6.4 for an implementation). Furthermore, figured bass serves as a guide for performance, especially for the instruments improvising the basso continuo accompaniment (e.g., harpsichord, organ, and lute). It can also serve pedagogical and theoretical purposes, which not only provides contrapuntal information on how to conduct the resolution of dissonances, also is a useful analytical tool for studying Baroque compositional and performance practices.

3.1.2.2 Harmony in the eighteenth century

In the eighteenth century, different harmony theories were proposed, and a significant one is by the French theorist and composer Jean-Philippe Rameau (1683–1764), which appeared first in the *Traité de l'harmonie* (Rameau 1722, quoted in Lester (2002, 759)) and introduced fundamental concepts of harmony we know today, such as chordal inversions, chord qualities (triads and seventh chords), and chord progressions. Specifically:

- Chordal inversion: Although chordal inversion was understood as a rule of thumb among some theorists and musicians in the late seventeenth and early eighteenth century (Lester 2002, 755), Rameau formalized the theory of chordal inversion. For example, he explained that chords such as 5/3, 6/3, and 6/4 triads containing the same notes were essentially the same chord in different positions, and further that 6/3 and 6/4 triads were different inversions of a root-position 5/3 triad (Lester 1994, 101–102).

[10] In this context, inversions refer to the chords whose bass note (the lowest note) is not the chord root.

- Chord types: Rameau believed that there were only two chord types, triads and seventh chords. Seventh chords were also considered as dissonant harmony (Lester 1994, 100), since they contain at least one dissonance, which is the chordal seventh.[11] Triads consist of a perfect fifth, which can be either a major or minor triad as defined today, and the seventh chords are built with an additional third interval on top of a triad. For a dominant-seventh chord, it contains a second dissonance, which is the diminished fifth between the third (leading tone) and the chordal seventh. Since the chordal seventh and the leading tone both create dissonances, Rameau suggested that the chordal seventh should resolve downwards and the leading tone should resolve upwards (Lester 1994, 107), which greatly resembles the voice-leading rules we know today for dominant seventh chords.
- *Fundamental bass* and harmonic progressions: The term *fundamental bass* was invented by Rameau to denote the imaginary bass line which is produced by assembling the roots of chords as a progression; it is different from the bass line that is actually sounding, where chords may be presented in inversions (Lester 1994, 105). According to Lester (2002, 761), Rameau saw the fundamental bass as the harmony generator, and he realized that chord roots could be the basis of a powerful explanation of harmonic progression.

Rameau's theory of fundamental bass "spread rapidly throughout Europe" (Lester 2002, 772) and "the practical value of the fundamental bass for teaching composition and analysis proved attractive to musicians across Europe" (Lester 2002, 773). Lester concludes that "eighteenth-century harmonic theory—largely Rameauian theory and its legacy—transformed earlier thinking about how pitches interacted, and laid the groundwork for the conceptualizations about harmony, voice-leading, and the forces that give directionality to tonal music" (Lester 2002, 774).

3.1.2.3 Harmony in the nineteenth century

In the nineteenth century, harmony became more diverse and complex, largely through *chromaticism*, which refers to the use of notes outside the major or minor scales. Compared to the seventeenth and eighteenth century, its use increased significantly in the nineteenth century, where

[11] From Rameau's *Traité de l'harmonie* of 1722, Book 3, Chapter 39.

"*chromaticism* increased to the point that the major-minor key system began to be threatened" (Benward and Saker 2003, 46).

Additionally, the Roman numeral notation[12] was proposed by the German composer and theorist Georg Joseph Vogler (1749–1814), and he used Roman numerals consistently in his treatises (Bernstein 2002, 780). However, he used only capital Roman numerals to represent chords, where the quality of chords is not indicated (see Figure 3-1).

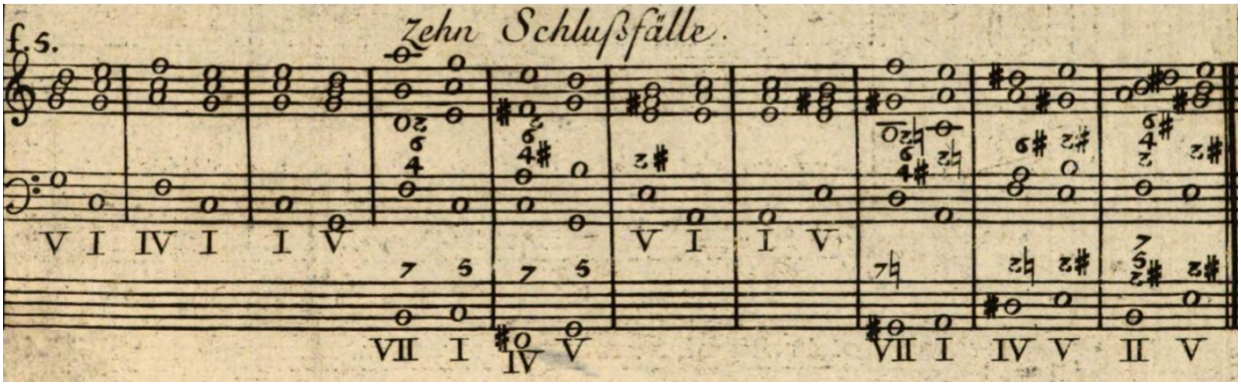


Figure 3-1: Example of Volger's use of Roman numerals in Gründe der Kuhrpfälzischen Tonschule in Beispielen, Vogler (1776), Table XXI, Fig. 5.

Another German composer, Gottfried Weber (1779–1839), further developed the Roman numeral notation (Bernstein 2002, 783). In his treatise *Versuch einer geordneten Theorie der Tonsetzkunst* (Theory of Musical Composition, 1817–1821), he introduced the larger numerals for major chords and smaller numerals for minor chords, "º" for diminished chords, and slashed 7 for major sevenths, as shown in Figure 3-2. Apparently, "Weber's roman numeral notation system achieved widespread popularity in the second half of the nineteenth century" (Bernstein 2002, 787).

[12] In this notation, each chord label is represented as a Roman numeral scale degree within the current key.

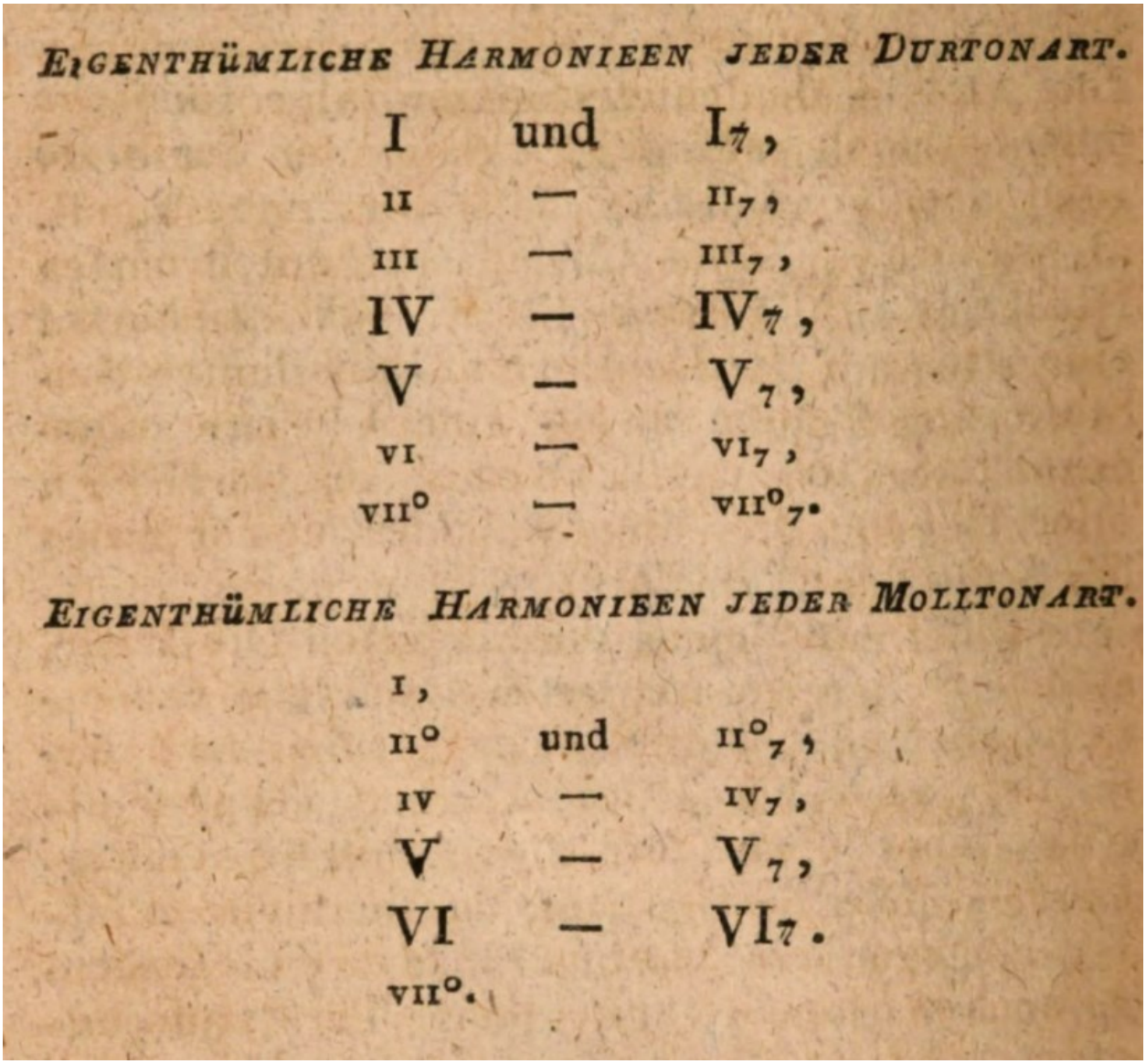


Figure 3-2: Gottfried Weber's illustration of Roman numerals applied to the scale degrees of the major and minor modes, triads shown on the left and sevenths shown at the right. "°" indicate diminished fifths, dashed sevenths indicate major seventh chords, sevenths alone indicate dominant-seventh chords, and the combination of ° and sevenths indicate half-diminished seventh chords. From Versuch einer geordneten Theorie der Tonsetzkunst, Vol. I, Book 2, p. 258.

Another theory of harmony proposed in this century—harmonic function—was by the German theorist Bernard Riemann (1826–1866) (Bernstein 2002, 796). Instead of focusing on a chord's relationship to the tonic, this theory concerns the function and purpose of a chord, not its identity (Selway et al. 2020). Riemann proposed three types of functions for harmony: tonic,

dominant, and subdominant, where the upper-case letters (`T, S, D`) and lower-case letters (`t, s, d`) are respectively used for the functions in major chords and minor chords (Hyer 2011). Chords with the same function can be considered as substitutes for one another. According to Riemann, there are three basic substitutions that transform one chord to another: *Variante, Parallele, and Leittonsweschel* (Hyer 2011; Selway et al. 2020), as shown in Figure 3-3:

- *Variante* describes the substitution between a major triad and a minor triad, where the major third can move down by a semitone to turn a major triad into a minor triad, or up a semitone for minor to major. Figure 3-3(a) shows an example: C minor (`t`) and C major (`T`) are related by a *Variante* substitution.
- *Parallele* is the substitution that connects a relative pair of major and minor triads, where the chord roots are a minor third apart. This is achieved by either moving a fifth of the major triad up a whole tone or moving the root of the minor triad down a whole tone. In Figure 3-3(b), C major (`T`) can move to A minor (`Tp`) if the fifth (G) is moved up a whole tone, forming the root of A minor. In reverse, the root of A minor can move down a whole tone to form the C major chord.
- *Leittonsweschel* connects a pair of major and minor triads whose roots are a major third apart (e.g., C major and E minor). In this substitution, the root of the major chord can move down by a semitone, or the fifth of the minor chord can move up by a semitone (see Figure 3-3(c)).

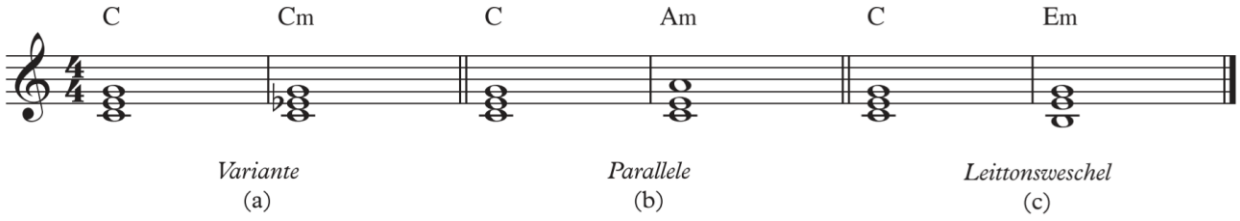


Figure 3-3: Illustration of the three basic substitutions of Riemannian theory, Variante (a), Parallele (b) and Leittonsweschel (c), using C major triad as an example (Selway et al. 2020, 140).

Each function can also be associated with multiple chords, as shown in Table 3-1. Table 3-1 also shows how chords with the same function can be substituted for one another. Through these substitutions, we can see that each chord contains at least one function, and some chords, such as `vi` and `iii`, can contain more than one function (Selway et al. 2020).

Table 3-1: Chords that share one of the three harmonic functions through different substitutions: Variante, Parallele (P or p) and *Leittonsweschel (L or l). Upper-case letters indicate substitutions from minor triads to major triads, and vice versa for lower-case letters. The first letter in parentheses indicates the harmonic function: Tonic (T or t), subdominant (S or s), and dominant (D or d). The second letter (when applicable) indicates how the chord can be obtained with substitutions. For example, iii (Tl) chord can be obtained from I (T) chord using Leittonsweschel*, and *♭iii chord (tP) can be obtained from i (t) chord using Parallele. Table from (Selway et al. 2020).*

Function	**Chord (substitution)**
Tonic	I (T), iii (Tl), vi (Tp), i (t), ♭iii (tP), ♭VI (tL)
Subdominant	IV (S), ii (Sp), vi (Sl), iv (s), ♭II (sL), ♭VI (sP)
Dominant	V (D), iii (Dp), vii (Dl), v (d), ♭III (dL), ♭VII (dP)

Bernstein (2002, 800) states that Riemann's "theory of functionality became widely adopted throughout Europe and, indeed, is still clearly to be seen in harmony textbooks in Germany, Scandinavia, and Russia today." His theory also offers a basis for measuring chord distance and similarity, which has been applied to the computational study of harmony (Chuan and Chew 2008; Selway et al. 2020).

In the nineteenth century, a new chord label representation—*chord letters*—also emerged. Chord letters are known to represent chords as the combination of chord root and chord quality using letters of the alphabet and other symbols. For example, "Arthur von Oettingen used letters of the alphabet with a superscript '+' sign for major chords and a superscript zero for minor chords;" (Oettingen 1866, translated by Bent et al. (2001)). Riemann also referred to the meaning of chord letters in *Vereinfachte Harmonielehre (Harmony Simplified)*, saying: *"*In these examples, the harmonies are not indicated as tonic, dominant, and subdominant, but by small italic letters (c, ^{o}e) indicating their principal notes." (Riemann 1896, 24).

In the twentieth century, the representation of chord letters further developed and was adopted for guitar, keyboard, and other instruments (Bent et al. 2001). Chords could be identified with a letter for the root, 'm' for a minor chord quality, superscript zero for a diminished chord

quality, and '+' for an augmented chord quality; the letter itself can represent a major chord quality (Bent et al. 2001).

3.2 Chord label representations

As discussed in Section 3.1, chord labelling has been used in composition, pedagogy, and music analysis since the common practice period, with different representations of chord labels proposed. In this section, I will introduce three chord label representations—figured bass, Roman numerals, and chord letters—in detail.

3.2.1 Figured bass

Figured bass is arguably the first chord label representation that has been widely used in Western music, spanning almost two hundred years from the early seventeenth century to the late eighteenth century. It is a type of music notation that uses numerals and other symbols to indicate intervals to be played above a bass note (Williams and Ledbetter 2001). Figure 3-4 shows an example of figured bass, as well as how a harpsichordist might realize the figured bass as an improvised accompaniment. Such realizations are not typically explicitly included in scores, as the musical tradition of the time left them to be improvised based on the skills and taste of the continuo player. Additionally, there are three aspects of figured bass annotations (FBAs) that should be highlighted (Chew and Rastall 2014):

- Numbers with slashes through them indicate altered intervals. Backward slashes usually indicate raised intervals (e.g., m. 3.3 and m. 3.4 in Figure 3-4), and forward slashes usually indicate lowered intervals (e.g., 𝄫).
- FBAs followed by continuation lines indicate that the harmony of the preceding figure is prolonged (e.g., m. 1.4, m. 4.2, and m. 4.4 in Figure 3-4).
- Multiple FBAs over a stationary bass (e.g., 4–3 in m. 5 in Figure 3-4) usually indicate a suspension being resolved.

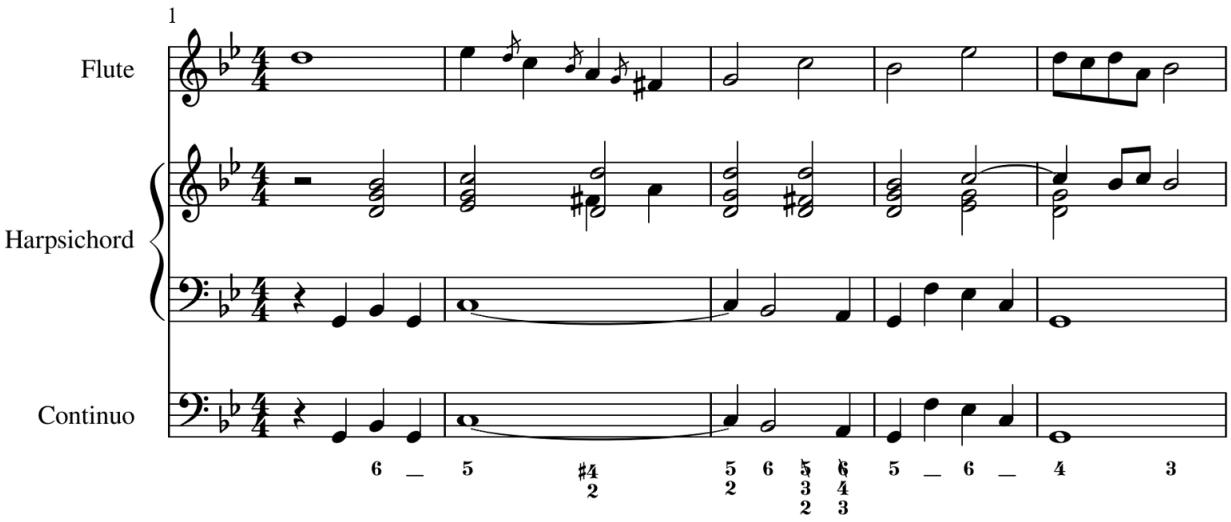


Figure 3-4: A sample musical passage, where figured bass annotations (FBAs) are shown below the continuo line, and where the added harpsichord line is an example of what a continuo player might improvise based on the figured bass. Figures indicate intervals above the continuo line that could be played in the improvisation. For example, the "6" in the first measure corresponds to the pitch class "G", which is a 6th above the bass "B♭". An actual improvisation would likely also typically contain the pitch class "D" (a 3rd above the bass "B♭") in this slice, but this is not explicitly indicated in the figures. This is an example of how FBAs do not always specify all the notes to be played by the continuo player, and usually omit some obvious figures (Williams and Ledbetter 2001).

Figured bass also has the following characteristics, following the discussion from Section 3.1.2.1:

- Chord qualities are not explicitly specified. For example, the figure "7" can indicate any kind of seventh chords—dominant 7th, minor 7th, major 7th, half-diminished 7th, or fully diminished 7th. One can only clarify this using the key signature and the scale degree of the bass. For example, the figure "7" over the bass "B" can be a dominant 7th chord in the key of E major, a minor 7th chord in D major, a major 7th chord in F♯ major, a half-diminished 7th chord in A (natural) minor, and fully diminished 7th in A (harmonic) minor.
- The chord root is also not specified and must be inferred by considering the bass and the chordal inversion the figured bass indicates.

Without specifying the chord root and chord quality in the notation, perhaps figured bass made it harder to quickly grasp the specific harmonic content of the composition, compared to Roman numerals and chord letters introduced below.

3.2.2 Roman numerals

As discussed in Section 3.1.2.3, Roman numerals were first used consistently by Volger (Bernstein 2002, 780), then popularized by Weber, and are still used in modern music theory and analysis. An example of the modern Roman numeral syntax used by Benward and Saker (2003) is shown in Figure 3-5.[13]

Modern Roman numerals explicitly show the chord quality using upper-case numerals for major chords, lower-case for minor, and superscripts for other chord qualities and inversions. Furthermore, it requires analysts to provide the key information for the analyzed passage and specify the new key whenever there is a modulation. The actual chord root (pitch-class name) can be inferred using Roman numerals and the key information (i.e., the chord root for the first Roman numeral "I" in Figure 3-5 is "A" since it is the tonic chord of A major key).

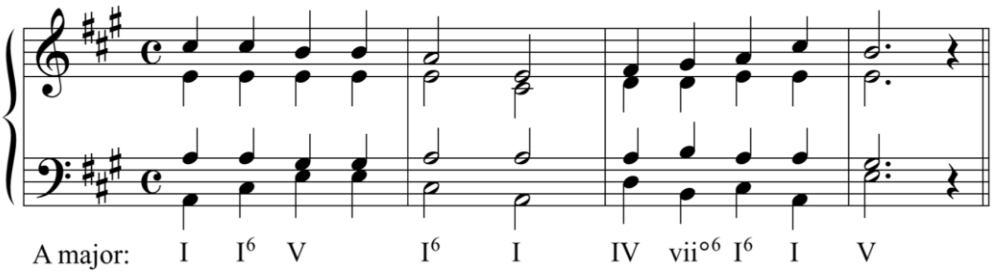


Figure 3-5: Roman numeral analysis for "St. Lucian" by Christian H. Rinck for measures 1 through 4. Image source: Music in Theory and Practice (Benward and Saker 2003), Vol. 1, Fig. 4.24, p. 83.

3.2.3 Chord letters

As introduced in Section 3.1.2.3, chord letters explicitly show both the chord quality and the chord root in the notation, but the key information is not specified, compared to Roman numerals. Chord letters are widely used in popular music and jazz, and the role of chord letters is more tailored for performance purposes, where musicians often sight-read; this notation therefore requires the minimum amount of time for musicians to figure out which notes to play (Harte et al. 2005). The notation of chord letters comes in various forms, where the chord quality can be

[13] The modern syntax uses superscripts, which are attached on the right-top side of Roman numerals, to indicate inversions and some chord qualities.

represented in multiple ways. For example, C major 7th chord can be noted as CM7, CMaj7, or C $^{\triangle 7}$ as needed (Harte et al. 2005). One example of chord letter notation is shown in Figure *3-6*. The information of chordal inversion is optional, and can be added with a forward slash plus the pitch class name of the bass if specified (e.g., "F♯o" will be "F♯o/A" in the first inversion).

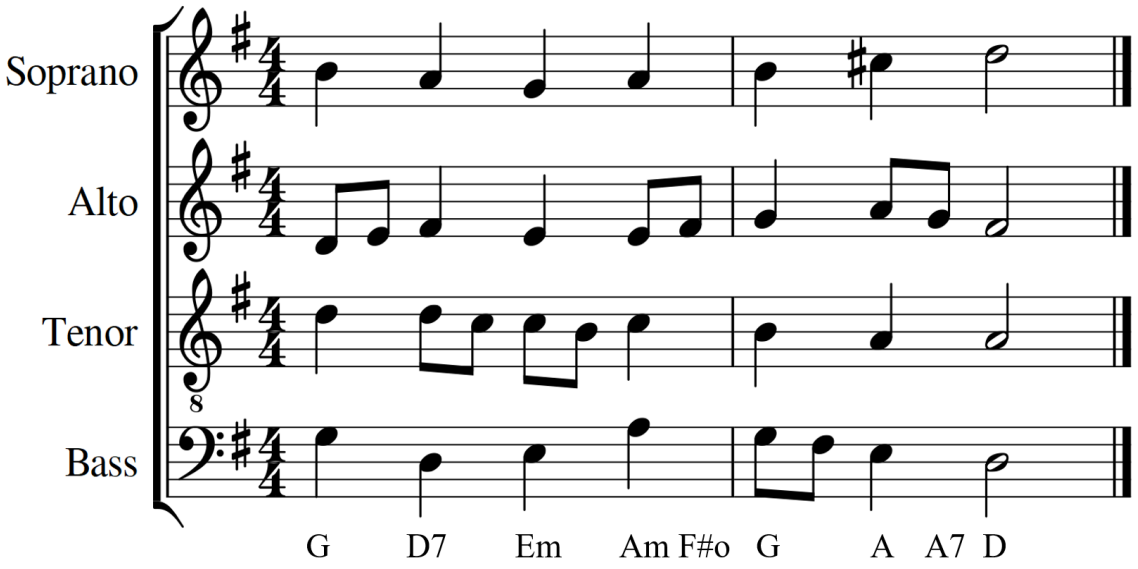


Figure 3-6: An example of chord letter notation for a 4-voice chorale passage. Image modified from Ju et al. (2019, 862).

3.2.4 Summary

A comparison of the three representations for chord labels is shown in Table 3-2 and Figure 3-7. Overall, Roman numerals and chord letters are commonly used in modern music analysis, and figured bass is seen more as a notation mainly for the Baroque music. Nonetheless, figured bass provides an early indication of harmony and can be used to obtain chord labels for many Baroque repertoires (see implementation in Section 6.4), and may offer meaningful insights into a composer's unique compositional style and can be of interest to both music-theoretical and musicological study.

Table 3-2: A summary of pros and cons for Figured bass, Roman numerals, and chord letters representing chord labels.

Notation name	**Pros**	**Cons**
Figured bass	Voice-leading information indicated (e.g., 8–7), especially the resolution of suspensions (e.g., 4–3).	No chord roots or chord qualities are specified.
Roman numerals	Indicate both chord qualities and key information explicitly.	Chord roots (pitch class names) are not explicit and have to be inferred using both Roman numeral and key.
Chord letters	Both chord qualities and chord roots (pitch class names) are indicated explicitly.	No key-related information is indicated.

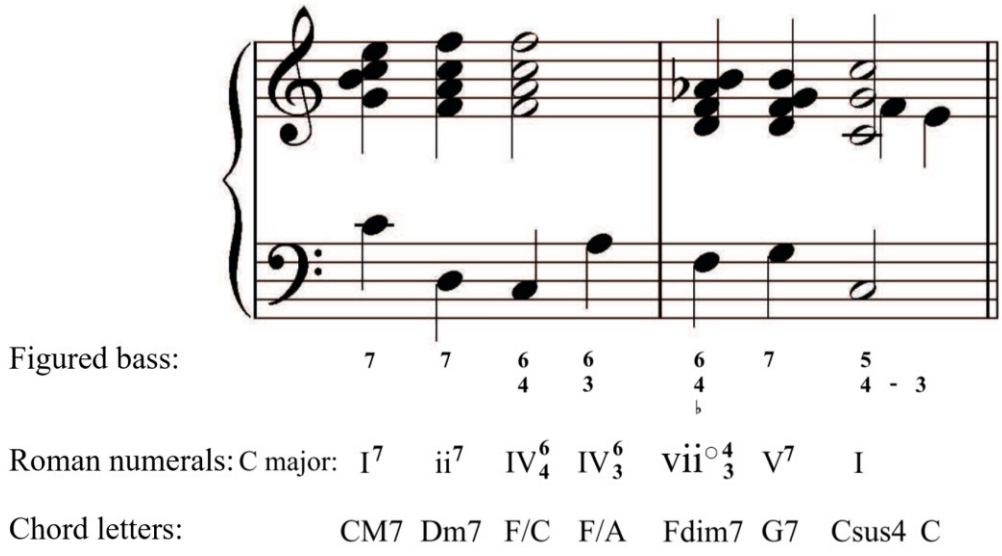


Figure 3-7: Summary of three different representation examples of chord labels discussed in Section 3.2, from top to bottom: Figured bass, Roman numerals, and chord letters. Example modified from Harte et al. (2005, 67).

3.3 All roads lead to Rome: Different approaches of chord labelling

In Section 3.1, a brief history of harmony is introduced with perspectives on harmony from a few prominent theorists. It is fair to say that these approaches to chord labelling differ,[14] which can potentially result in a number of alternative analyses for the same musical passage, creating a great degree of analytical ambiguity—a fundamental problem which this dissertation aims to address. In this section, greater details of chord labelling by three prominent music theorists: Rameau, Weber, and Schenker, will be introduced.

3.3.1 Jean-Philippe Rameau (1683–1764)

Rameau's approach to chord labelling still has a great influence today. In this section, an analysis of his own motet—*Laboravi Clamans* from his *Traité de l'harmonie*—will be used as an example of his chord labelling method (from Beach (1974, 278), see Figure 3-8).

Based on his theory of chordal inversion (see Section 3.1.2.2 for details), Rameau first derived fundamental bass, which indicates the root of a chord, shown as the bottom voice of Figure 3-8. Although triads and 7th chords are the only chord qualities he considered, other dissonances such as fourths and ninths, can be converted to the sevenths and incorporated as part of the harmony, according to his theory of supposition (Lester 2002, 764). Take m. 12.1 of Figure 3-8 as an example, Rameau argued that the ninth chord built on D (D-F-A-E) is essentially a 7th chord on F (F-A-E), with D "supposed" a third below the fundamental bass (Beach 1974, 282). As a result, the ninth becomes part of the derived 7th chord and the eighth from the 9–8 suspension forms a subsequent triad, resulting in two separate chord labels. The same trend of interpreting suspension as two separate chord labels can also be found at m. 10 of Figure 3-8, where the seventh and the sixth of the 7–6 suspension both form a different chord on their own.

[14] Although these approaches adopted different chord label representations, it is the underlying methods of labelling chords that lead to the essential ambiguities, as discussed in Section 3.3.4.

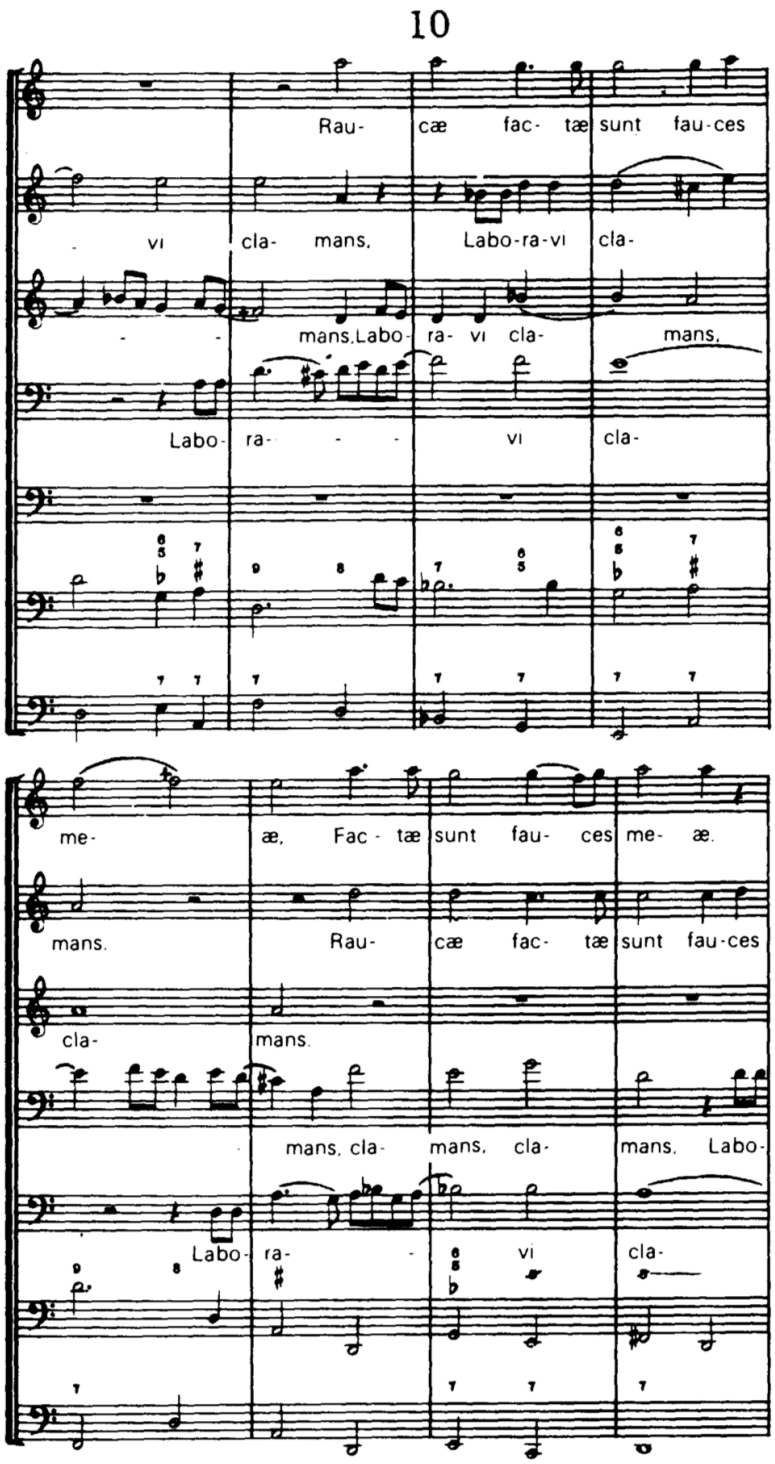


Figure 3-8 Rameau's analysis for measures 8 through 15 of his motet Laboravi Clamans, which contains the fundamental bass voice at the bottom. Each fundamental bass note indicates a chord, where the unfigured ones and the ones with figure "7" indicate triads and seventh chords, respectively. The example is taken from Beach (1974, 278).

Rameau's interpretation of suspensions is a good example of his conception of harmony, where chord labelling is mostly done in a vertical manner, and the voice-leading context is rarely considered (Beach 1974, 282). Overall, his approach of chord labelling is fairly local and granular, which incorporates most dissonances as chord tones with his theory of supposition, resulting in many 7th chords and frequent chord changes.

3.3.2 Gottfried Weber (1779–1839)

The German composer and music theorist Gottfried Weber includes his methods of chord labelling in the first volume of his treatise *Versuch einer geordneten Theorie der Tonsetz-kunst* (Theory of Musical Composition, 1817–1821). He seems to have popularized Roman numerals as an alternate system to represent chord labels (Bernstein 2002, 787). In the last two volumes of Weber's treatises, he used Roman numerals to present his analysis of chord labelling for excerpts of different compositions (Beach 1974, 298). One example is the analysis for *The March of the Priests of Isis* from Mozart's *Magic Flute*, shown in Figure 3-9. Beach (1974, 298) notes that "Weber's conception of harmony is much more like Kirnberger's than Rameau's. Note for example his interpretation of the 9-8 suspension in m. 6 (or the 4-3 in m. 12) of Figure 3-9 as representing a single harmony. However, like both[15] theorists, he shows a tendency to interpret too many chords as fundamental harmonies."

Apart from suspensions, Weber identified other types of what he described as "transition-tones" (Durchgehende Töne, translated by Saslaw (1992, 239)), such as appoggiaturas, passing tones, and neighbour tones (see Glossary). When appropriate, these tones were identified as non-chord tones (Saslaw 1992, 248), especially if the interpretation of harmony contains many chord changes within a bar, especially on unaccented beats and notes with shorter durations (Saslaw 1992, 247), as shown in Figure 3-10.

[15] Kirnberger and Rameau.

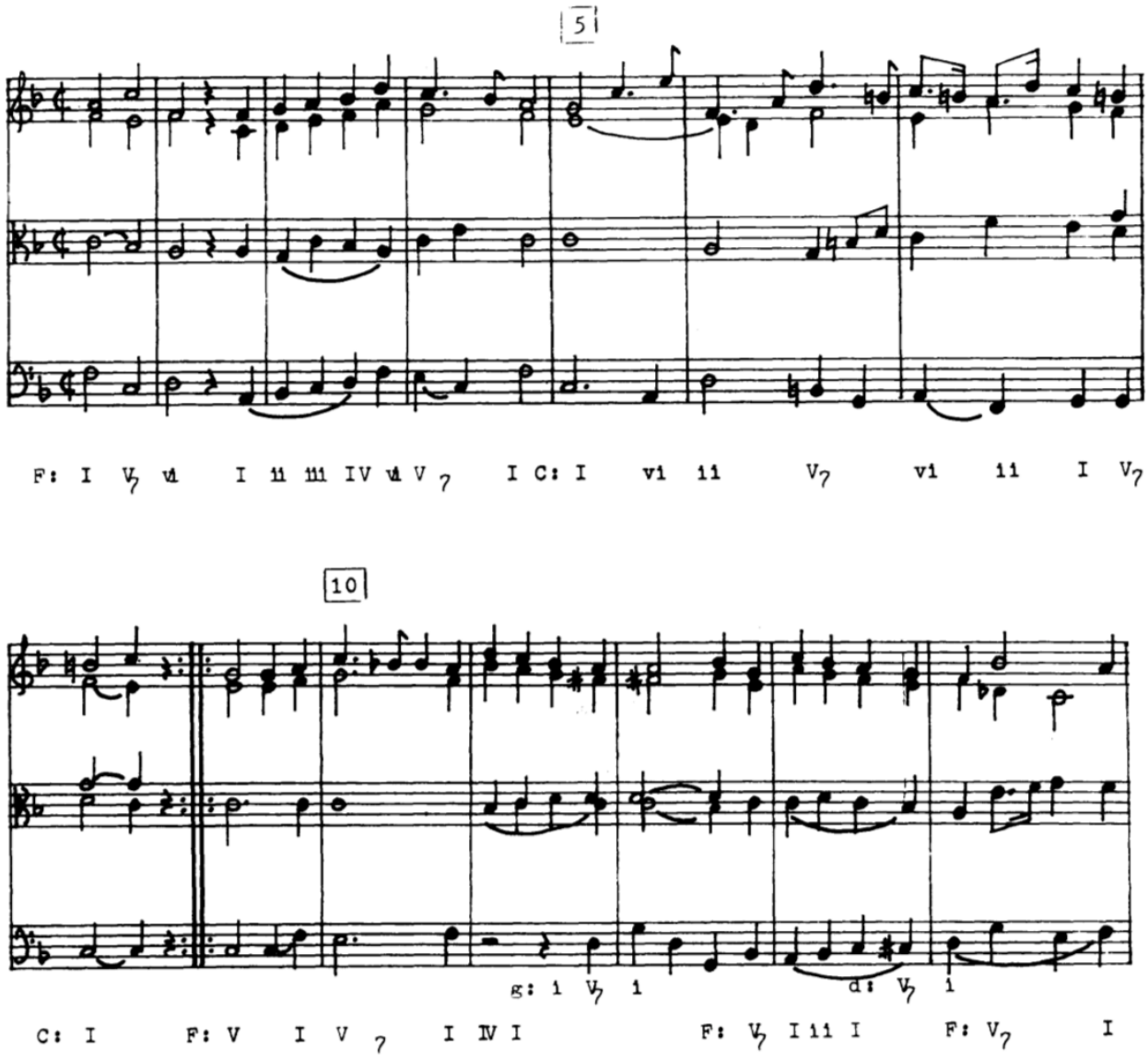


Figure 3-9: Weber's Analysis for measure 1 through 14 of The March of the Priests of Isis *from* Mozart's Magic Flute *(Beach 1974, 300).*

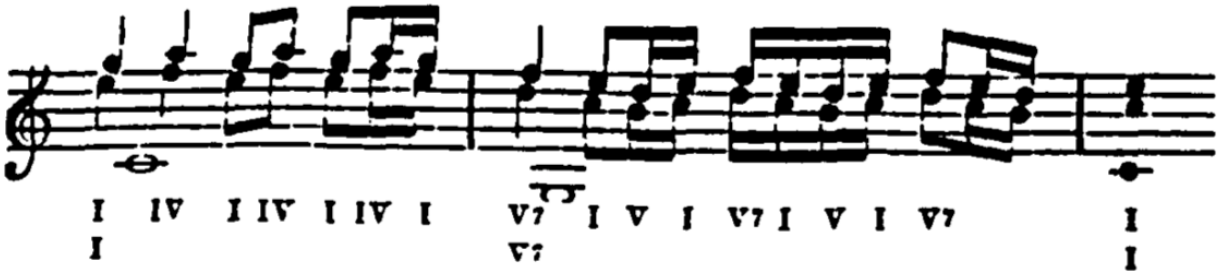


Figure 3-10: An example of a musical passage either considering every simultaneity as harmonic, leading to more labels (upper row), or considering some as non-chord tones using voice-leading, resulting in fewer chord labels (lower row), according to Weber (Saslaw 1992, 248).

Although Weber made conscious decisions of identifying non-chord tones and excluding them from harmony, his analysis can still be considered as local and granular, showing "a tendency to interpret too many chords as fundamental harmonies" (Beach 1974, 298). Note his analysis of the parallel sixth chords in m. 3 of Figure 3-9, where he assigned a Roman numeral to every chord, implying that each is a separate harmony. Alternatively, the chords in m. 3 can be considered as the extension of a single harmony (the *ii* chord) "as preparation for the following dominant" (Beach 1974, 298), (the *V* chord) at m. 4.

According to Beach (1974, 298), "this analysis also demonstrates Weber's method of relating all harmonies to their proper tonal center", which contains frequent, short modulations[16] that "involve only two or three chords" (Beach 1974, 298), such as the G minor and D minor modulation shown at m. 11–12 and m. 13–14 of Figure 3-9 , respectively. In modern music theory, these short modulations may be interpreted as tonicizations.

[16] Change from one key to another.

3.3.3 Heinrich Schenker (1868–1935)

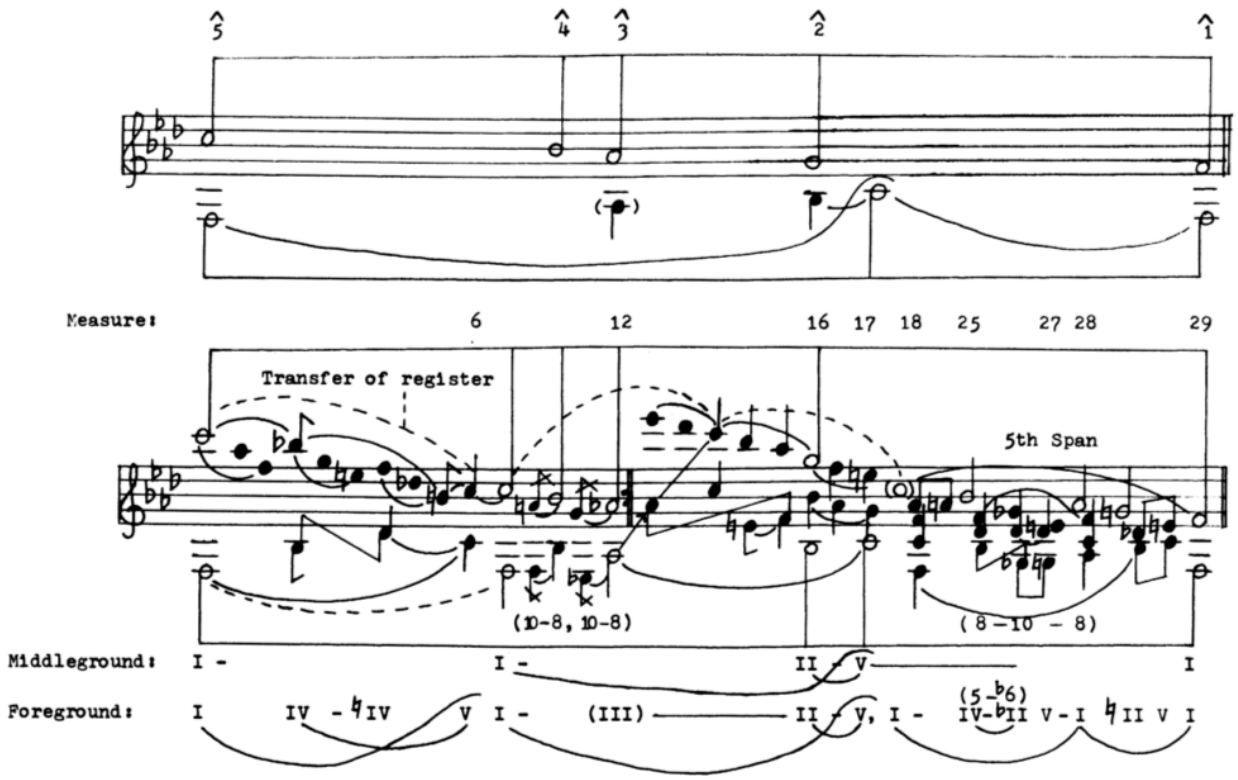


Figure 3-11: Schenker's analysis of the theme from Hadyn's Andante con variazioni in F minor*, measure 1 through measure 29 (Beach 1974, 302).*

According to Drabkin (2002, 816), Schenker believed that music is mainly tonal and highly hierarchical, where a composition is primarily governed by the principle chord, the tonic triad, and all other chords function as subordinates to the tonic. His analysis of music almost always made a distinction between passing and essential harmonies. Likewise, the notes from a melody can also be described as transitional or essential (Drabkin 2002, 816). Heinrich Schenker's method of music analysis (known as "Schenkerian analysis") is often seen as a reductive method, combining the component elements of melody, harmony, counterpoint, and form into an all-encompassing analysis at different inter-related levels (Beach 1974, 303). An example is shown in Figure 3-11, where three different layers can be abstracted from the music. The first layer and second layer of analysis are called *foreground* and *middleground*, respectively, and the corresponding Roman numerals are indicated below the lower staff. The foreground layer is the reduction of the original

music, with non-chord tones identified and eliminated from the musical surface. Therefore, the resulting chord labels reflect an abstraction of the harmonic content, with considerations of voice-leading; the chord labels in the middle ground layer only show the harmonies that are structurally important and have been "prolonged" in the composition, with all the intermediate chord labels are omitted and only highlight the larger structures of the piece; the third layer of analysis—known as the *background layer*—is shown in the upper staff of Figure 3-11, which only shows the skeletal structure of the composition (Beach 1974, 303).

Although Schenkerian analysis is not exactly about chord labelling, the idea of reducing the musical surface to different layers may potentially explain a primary source of chord labelling ambiguity, where some annotators (e.g., Rameau) may conduct chord labelling directly on the musical surface, leading to more granular analysis, and others (e.g., Weber or Schenker) may reduce the musical surface to a certain degree using voice-leading and other musical concepts, resulting in a more abstract analysis.

3.3.4 Summary

Chord labelling is known as a subjective endeavor (Koops et al. 2019), and usually we cannot pinpoint a single correct analysis. Section 3.2 introduced the main principles of chord labelling by three prominent music theorists: Rameau, Weber, and Schenker, where each theory has an audience of its own. These variant approaches to chord labelling can be largely understood as balancing chord tones and non-chord tones in the tonal context, where Rameau's theory of supposition incorporated more dissonances as chord tones, compared to Weber who considered voice-leading to classify the dissonances in transitional motions (see Glossary) as non-chord tones. Schenker, on the other hand, proposed multiple layers of chord labelling, each with a different level of details and analytical perspective in mind.

3.4 Automatic chord labelling: A computational approach

Despite being an essential tool for music analysis, chord labelling is a complex process, which is time-consuming and requires years of training. Automatic chord labelling presents a promising alternative for obtaining chord labels in an inexpensive way. Although it exclusively concerns symbolic music in this dissertation, chord labelling can also be applied to audio signals,

and this task is often referred to as *automatic chord estimation* (Humphrey and Bello 2015). Although these are two different tasks, they are still related, including the fact that ambiguity in both tasks has been acknowledged and discussed in the past.[17] However, they do have different methodologies on building automated models. For a detailed overview of automatic chord estimation, please refer to McVicar et al. (2014), and the remaining discussion of this section will focus exclusively on automatic chord labelling for symbolic music.

As discussed in existing reviews of automatic chord labelling (Mouton and Pachet 1995; Kröger, Passos, and Sampaio 2010; De Haas 2012; Mearns 2013; Rizo, Illescas, and Iñesta 2016), automated approaches have presented a promising alternative in recent years, and this section will introduce various automatic chord labellers proposed in the literature to date.

Typically, an automatic chord labeller will first divide music into smaller segments, from which relevant musical features are extracted as inputs to either a rule-based or a machine learning model, which outputs chord labels automatically. To assess the model's performance, evaluation metrics are used to compare the generated results against manual expert annotations. These aspects of automatic chord labelling will be introduced in the following subsections.

3.4.1 Segmentation

The first step for chord labelling is to divide music into smaller segments, where each either has fixed duration or varied durations, and which can be associated with a chord label.

3.4.1.1 Frames: Segments with a fixed duration

One way to segment the music is to divide it into fixed durations. Similar to audio chord recognition, which usually refers to identifying chord letters from an audio signal, the music is divided into a set of *frames*—each is considered as the minimal, meaningful unit with a fixed duration. People have used a wide variety of values, which can be the duration of a 32nd note (Chen and Su 2018; Micchi, Gotham, and Giraud 2020), an 8th note (Micchi, Gotham, and Giraud 2020), a musical beat (Mearns 2013), a half measure (Raphael and Stoddard 2004), or the shortest note

[17] For audio signals, see Humphrey and Bello (2015); Selway et al. (2020); for symbolic music, see Condit-Schultz, Ju, and Fujinaga (2018).

found in the music (Temperley and Sleator 1999; Barthélemy and Bonardi 2001; Illescas, Rizo, and Iñesta 2007).

3.4.1.2 Slices: Segments with varied durations

Slices in this dissertation are considered as segments with varying durations, which are often termed as "salami-slices" or "note onset slices", formed whenever a new note onset occurs in any musical voice, and each consists of all musical notes sounding at that moment (Pardo and Birmingham 2002; Radicioni and Esposito 2007; Kröger et al. 2008). For example, the music excerpt in Figure 3-12 contains 10 note onset slices, separated by the vertical dashed lines.

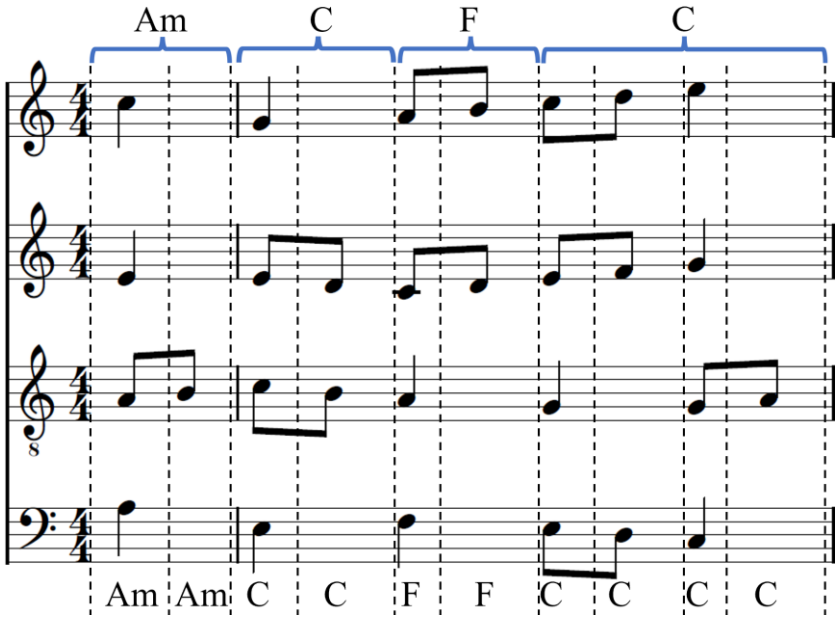


Figure 3-12: Illustration of two different approaches that obtain segments with varying durations, using chord segmentation algorithm (above, see Section 3.4.1.3) and note onset slice (below, see Section 3.4.1.2), based on the first measure (with pickup measure) of the Bach chorale BWV 33.06 "Allein zu dir, Herr Jesu Christ."

3.4.1.3 Chord segmentation

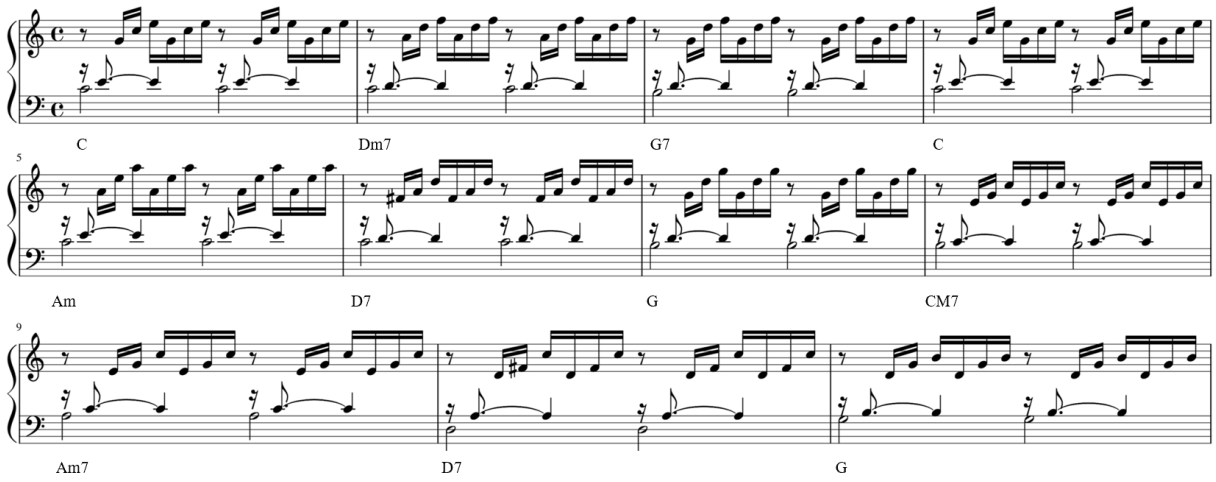


Figure 3-13: J. S. Bach Prelude in C, BWV 846, Measures 1–11 of the score with chord labels in the text below the lowest stave. Image from (Micchi, Gotham, and Giraud 2020).

Frames and slices introduced above can be understood as approaches to capture all possible chord changes. However, the number of actual chord changes is often smaller than the number of frames or slices, and this is especially true in a musical passage with arpeggiation, such as Figure 3-13, where each measure only contains a single chord but can be divided into many frames or slices. This is also true in chorale texture that is largely homorhythmic, as shown in Figure 3-12, where there are 10 slices but only contain four chord labels. To explicitly capture the chord changes, machine learning algorithms have been proposed to auto-segment the music into different chord regions, and associate each one with a chord label (Masada and Bunescu 2019; Chen and Su 2019).

3.4.2 Musical features

Symbolic music can be represented in different formats, such as **kern, MusicXML, MIDI, and the Music Encoding Initiative format (MEI). The majority of automatic chord labelling literature either used MIDI (Pardo and Birmingham 2002; Mearns 2013; Barthélemy and Bonardi 2001) or **kern (Devaney et al. 2015; López 2017; Masada and Bunescu 2017) as symbolic representation. Both are compact symbolic formats, but MIDI lacks some essential musical information, including pitch spelling and measure boundaries.

To associate each segment with a chord label, the pitch content is an essential feature, which can either be represented as pitch names (Winograd 1968; Chen and Su 2018; 2019; Masada and Bunescu 2019; Micchi, Gotham, and Giraud 2020), or pitch class names (Illescas, Rizo, and Iñesta 2007; Temperley and Sleator 1999; Raphael and Stoddard 2004; Scholz, Dantas, and Ramalho 2005; Taube 1999; Radicioni and Esposito 2007; Hoffman and Birmingham 2000; Pardo and Birmingham 2002; Tsui 2002; Kröger et al. 2008; Micchi, Gotham, and Giraud 2020) that omit the octave information and reduce the dimensions of input features. Some of the work mentioned above also considered pitch spelling (Temperley and Sleator 1999; Tsui 2002; Scholz, Dantas, and Ramalho 2005; Kröger et al. 2008; Micchi, Gotham, and Giraud 2020), which is important to determine the identity of some chords, such as German augmented 6^{th} chord (can be spelled as a dominant 7^{th} chord enharmonically), and fully diminished chords (can be spelled in four different ways).

Some work has separated the bass voice and considered its pitch content apart from the content of upper voices (Masada and Bunescu 2017; 2019; Chen and Su 2019; Micchi, Gotham, and Giraud 2020), and Micchi, Gotham, and Giraud (2020) further conducted comparative studies on both the addition of pitch spelling and the separation of the bass voice for automatic chord labelling, concluding that:

- The inclusion of the bass information results in significantly higher accuracies in determining chord root, chord quality, chordal inversion, and key.
- The inclusion of pitch spelling overall results in slightly higher accuracies overall, but the results are not significant statistically. The reason might be that although pitch spelling may improve the accuracy of identifying some chords, it inevitably expands the dimension of the input features significantly and may compromise performance, especially when the amount of training data is limited.

Besides pitch content, metrical context can also be considered, such as information on whether a segment is on a downbeat, a weak beat, or a fractional beat (Temperley and Sleator 1999; Tsui 2002; Radicioni and Esposito 2007; Masada and Bunescu 2017; 2019). This feature enables automatic chord labellers to identify segments with different beat strengths. Additionally, some work also includes context: Segments directly preceding and following the current segment are considered as part of the input features (Tsui 2002; Kröger et al. 2008; Chen and Su 2018; 2018;

Micchi, Gotham, and Giraud 2020). This context reflects the harmony theory that chord labelling is not an isolated process, and usually local context (e.g., counterpoint, voice-leading, and chord progression) is needed to associate a segment with a proper chord label.

Additionally, there are existing frameworks that can process symbolic music data, and some can extract the musical features mentioned above. For example, music21 (Cuthbert, Ariza, and Friedland 2011) is a Python-based, music analysis toolkit that can extract various kinds of musical features from symbolic music; such frameworks also include jSymbolic2 (McKay, Cumming, and Fujinaga 2018), the MIDI toolbox (Eerola and Toiviainen 2004), the Humdrum toolkit (Huron 2002), and the Melisma Music Analyzer (Temperley 2001), which could be used in the research of automatic chord labelling.

3.4.3 Non-chord tone identification

Identifying non-chord tones has been incorporated in many music analytical tasks, including melodic analysis (Illescas et al. 2011), polyphonic music retrieval (Pickens 2004), and harmonization (Chuan and Chew 2011). Its importance to automatic chord labelling has been acknowledged in the existing literature, and there are automated approaches that integrated non-chord tone identification as an explicit part of the methodologies (Hoffman and Birmingham 2000; Mearns 2013; Pardo and Birmingham 2002; Sapp 2007; Willingham 2013; Taube 1999). For example, Hoffman and Birmingham (2000) proposed a complete, independent algorithm that detects non-chord tones using voice-leading information. Once identified, these non-chord tones can be removed from the musical surface (see Figure 3-14), then the remaining chord tones can be mapped into chord labels.

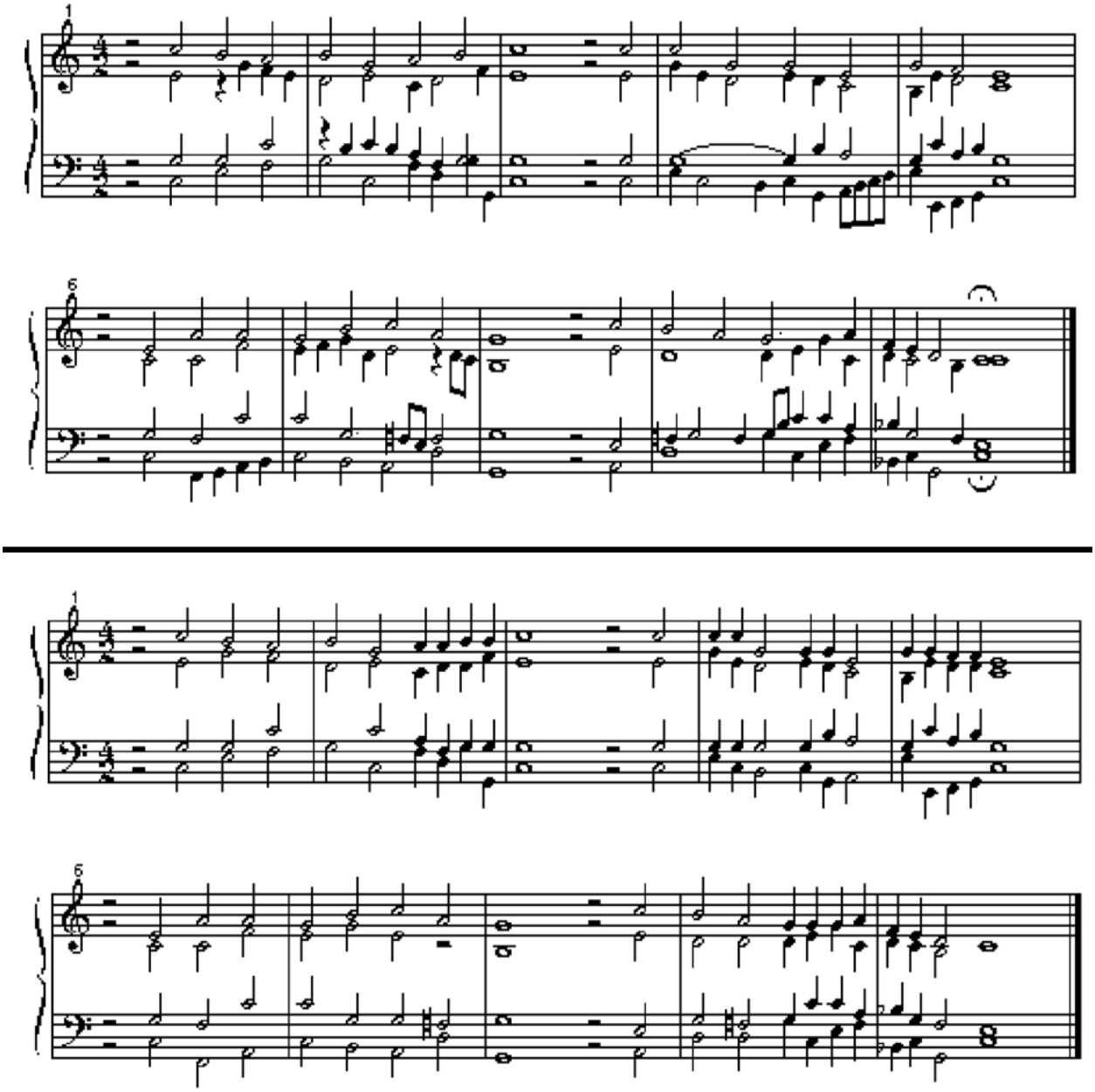


Figure 3-14: First 10 measures of Von Himmel Hoch, with the original score shown above (separated by the horizontal line) and the one without non-chord tones shown below (Hoffman and Birmingham 2000, 9).

3.4.4 Rule-based approach

The research of automatic chord labelling now has a history of more than 50 years, perhaps beginning with Winograd (1968), who proposed a rule-based algorithm using a combination of grammatical rules. Over the next 30 years, almost all automatic chord labellers were based on rules

(Temperley 1997; Temperley and Sleator 1999; Taube 1999; Temperley 2001; Maxwell 1992; Pachet 1991; Scholz, Dantas, and Ramalho 2005; Hoffman and Birmingham 2000; Tojo, Oka, and Nishida 2006; Rohrmeier 2006; de Haas 2012; Granroth-Wilding 2013; Barthélemy and Bonardi 2001; Pardo and Birmingham 2002). The reasoning for using a rules-based approach is obvious: Chord labelling is inherently a set of rules that associate the musical surface with chord labels. Starting in the seventeenth century, chord labelling has gradually become a common practice in composition, pedagogy, and music analysis, where composers and theorists proposed different sets of rules for chord labelling (see Section 3.3). Therefore, to build an automatic chord labeller is to codify these rules in a systematic order, and the main strength of this approach is its explainability: Each chord label is explicitly generated using a set of well-defined rules.

There are three main shortcomings of this approach for automatic chord labelling systems. First, such systems usually work by ordering a sequence of rules, and the propagation of errors from early steps may compromise the final result. Second, they often fail to produce correct analyses for even moderately exceptional passages, as it is extremely complicated to define comprehensive rules. Last, the rules are often carefully curated by experts for a specific genre of music or a particular composer, and such systems may not adapt to other music well.

3.4.5 Machine learning approach

With the rapid development of computational power and the availability of annotated data, the machine learning approach has become increasingly popular for automatic chord labelling in the past 20 years. Compared to a rule-based approach, machine learning models are capable of adapting to different music, since the chord labelling process is automatically learned from the provided training data.

Many machine learning techniques have been applied to automatic chord labelling, including hidden Markov models (HMM) (Raphael and Stoddard 2004; Mearns 2013), artificial neural networks (ANNs) (Tsui 2002; Kröger et al. 2008), conditional random field (CRF) (Masada and Bunescu 2017). Most recently, recurrent neural networks and attention mechanism approaches have been experimented with, such as bi-directional long-short term memory (BLSTM) (Chen and Su 2018), gated recurrent unit (GRU) (Micchi, Gotham, and Giraud 2020), and transformer (Chen and Su 2019). Although these models (e.g., GRU and BLSTM) are capable of capturing long-term

dependencies (Greff et al. 2017) between the musical surface and chord labels, they have many parameters and may demand a large quantity of data to train.

For example, Micchi, Gotham, and Giraud (2020) used GRU for automatic chord labelling and aggregated four datasets with Roman numerals: Theme and Variation Encodings with Roman Numerals (TAVERN) (Devaney et al. 2015), Annotated Beethoven Corpus (ABC) (Neuwirth et al. 2018), Beethoven Piano Sonata with Function Harmony (BPS-FH) (Chen and Su 2018), and Roman Text (Tymoczko et al. 2019), only amounting to about 73,000 chord label annotations, which is often significantly smaller compared to the amount of data available in natural language processing and speech recognition. This also indicates one of the main challenges of the machine learning approach: It requires a sufficient amount of manual annotations as training data, and in the case of chord labelling, obtaining such annotations is extremely expensive and time-consuming.

3.4.6 Chord label formats

The chord labels generated by the automated models can be in different formats, such as figured bass (Barthélemy and Bonardi 2001; Wead and Knopke 2007), Roman numerals (Illescas, Rizo, and Iñesta 2007; Winograd 1968; Raphael and Stoddard 2004; Scholz, Dantas, and Ramalho 2005; Taube 1999; Tsui 2002; De Haas 2012; Rohrmeier 2011; Chen and Su 2018; 2019; Micchi, Gotham, and Giraud 2020), or chord letters (Temperley and Sleator 1999; Radicioni and Esposito 2007; Pardo and Birmingham 2002; Mearns 2013; Kröger et al. 2008; Sapp 2007; Chen and Su 2018; 2019; Masada and Bunescu 2017; 2019).

3.4.7 Evaluation metrics

Most chord labelling work that includes quantitative results used classification accuracy as the evaluation metric (Pardo and Birmingham 2002; Tsui 2002; Mearns 2013; Kröger et al. 2008; Sapp 2007; De Haas 2012; Chen and Su 2018; 2019; Masada and Bunescu 2017; 2019; Micchi, Gotham, and Giraud 2020; Radicioni and Esposito 2007), which indicates the percentage of segments with predicted chord labels that are identical to the ground truth labels.

Regarding the results reported in each paper, it is safe to say that these automated systems have not reached or come close to the performance of human experts: Some systems reached ~80% accuracy for chord letters (Masada and Bunescu 2019), and Roman numeral analysis performs

worse, with only 42.8% accuracy for a combination of different genres of music in the common practice period (Micchi, Gotham, and Giraud 2020).[18]

Furthermore, the accuracy results are difficult to compare to each other, since different works often use different sets of music, and even if the music is the same, either the chord labels could be in different formats,[19] or the chord label annotations are different. In fact, only two papers were found with their results comparing against the existing literature: Mearns (2013) and Micchi, Gotham, and Giraud (2020). As of now, there seems no unanimous agreement on a single dataset as a benchmark.

3.4.8 Open-source chord label datasets

This section shows currently available open-source datasets with both symbolic music and chord labels consisting of Roman numeral or chord letters.[20]

- Theme and Variation Encodings with Roman Numerals (TAVERN) (Devaney et al. 2015): 27 themes and variations for piano (10 by Mozart and 17 by Beethoven) in **kern format, which contain 1,060 phrases. There are two versions of Roman numeral annotations, each prepared by a different analyst.[21]
- Annotated Beethoven Corpus (ABC) (Neuwirth et al. 2018): All string quartets by Beethoven (16 string quartets, 70 movements) in the MuseScore format, annotated with Roman numerals.[22]

[18] Although there have been better performances reported in the literature (Tsui 2002; Kröger et al. 2008), they reflect only one step of the chord labelling process, such as finding the chord root, determining the chord quality, or identifying the chordal inversion. Unfortunately, the overall performance of generating chord labels with these steps were not reported in these two papers.
[19] For example, both Masada and Bunescu (2019) and Micchi, Gotham, and Giraud (2020) used the TAVERN dataset (Devaney et al. 2015), but the former used chord letters, and the latter used Roman numerals as the chord label representation, therefore the results are not directly comparable to each other. (Unless Roman numerals are translated into chord letters or vice versa.)
[20] There is no open-source dataset of figure bass annotations prior to this research, where I presented a new Bach Chorales Figured Bass dataset in Section 6.2.
[21] https://github.com/jcdevaney/TAVERN
[22] https://github.com/DCMLab/ABC

- Roman Text (Tymoczko et al. 2019): Contains 24 preludes from the first book of Bach's Well-Tempered Clavier, and 48 romantic songs from France and Germany, annotated with Roman numerals.[23]
- Kostka-Payne Korpus: 46 excerpts accompanying the theory textbook *Tonal Harmony* (Kostka and Payne 1995), annotated with Roman numerals.[24]
- "Sun Quartets" (Nápoles López 2017): Op. 20, consisting of six string quartets from Joseph Haydn in **kern format, annotated with Roman numerals.[25]
- Beethoven Piano Sonata with Function Harmony (BPS-FH) (Chen and Su 2018): Consists of the first movements from 32 piano sonatas by Beethoven, annotated with both Roman numerals and chord letters.[26]
- The Rameau dataset (Kröger et al. 2008): 371 Bach chorales in Lilypond format, and 156 of them are annotated with chord letters.[27]
- The Craig Sapp dataset (Sapp 2007): 370 Bach chorales in **kern format, and 75 of them are annotated with Roman numerals.[28]
- Band-in-a-Box Jazz standards (Choi, Fazekas, and Sandler 2016): 2,486 Jazz songs annotated with chord letters in Jazz standards, available in the format of Band-in-a-Box.[29]
- Weimar Jazz Database: 456 songs with scores and chord letters in Jazz standards.[30]

3.5 Conclusion

This chapter introduced various aspects of chord labelling, including a brief history of Western harmony, chord label representations, and examples of three different approaches of chord labelling proposed by prominent music theorists (Rameau, Weber, and Schenker).

[23] https://github.com/MarkGotham/When-in-Rome
[24] http://davidtemperley.com/kp-stats/
[25] https://github.com/napulen/haydn_op20_harm
[26] https://github.com/Tsung-Ping/functional-harmony
[27] It was originally available at: https://github.com/kroger/rameau/tree/master/rameau-deps/genos-corpus but it is no longer available.
[28] https://verovio.humdrum.org/, under "Scores" -> "75 chorales w/ harm. analyses."
[29] http://bhs.minor9.com/
[30] https://jazzomat.hfm-weimar.de/dbformat/dbcontent.html

Additionally, the process of automatic chord labelling was discussed, and a literature review of existing work was provided. In Chapter 4, I will introduce music theory treatises that discuss the analytical ambiguities in chord labelling, computational studies of variance in chord labelling among different annotators, and how this issue can be addressed in automatic chord labelling.

Chapter 4 Ambiguity in chord labelling: A case study

In Section 3.3, I introduced three different approaches to chord labelling, proposed by Rameau, Weber, and Schenker. The availability of multiple legitimate approaches to chord labelling reveals its ambiguous nature, where the chord labels prepared by different experts or even the same expert can vary (Koops et al. 2019; Condit-Schultz, Ju, and Fujinaga 2018). To have a systematic understanding of chord labelling ambiguity and how it can be addressed in automatic chord labelling, I first reviewed the main approaches that address ambiguity in machine learning in Chapter 2; this chapter specifically reviews the existing literature on chord labelling ambiguity in music theory (Section 4.1) and music information retrieval (MIR, see Section 4.2). In the end, methodologies for addressing ambiguity in automatic chord labelling will be proposed in Section 4.3.

4.1 Chord labelling ambiguity in music theory

According to Saslaw (1992, 40), the concept of chord labelling ambiguity was mentioned by the French composer and theorist Jean-Philippe Rameau (1683–1764), where he used the term *double employ* (double deployment) in the phrase "un double emploi à l'accord de la Seconde" (a double employment of the Chord of the Second) to refer to chords with two identities. For example, the pitch class set C-D-F-A can be interpreted either as an inversion of a subdominant, in the key of C major, with an added sixth (i.e., F*add6* chord, F-A-C-D) which may progress to the tonic chord, or as an inversion of a seventh chord (i.e., Dm7 chord, D-F-A-C), which can be followed by the dominant chord (Saslaw 1992, 41).

In the late 18th century, the German music theorist Georg Joseph Vogler (1749–1814) used the term *mehrdeutigkeit* (multiple meanings) to refer to music, including chords (Saslaw 1992, 47–52). According to Saslaw (1992, 52), he began with an example that pitch-class names (e.g., "C") common in different keys (e.g., C major, A minor, and G major) can have different scale-degrees (e.g., $\hat{1}$ in C major, $\hat{3}$ in A major, and $\hat{4}$ in G major). Similarly, the same chord letter (e.g., Am) could serve several functions in different keys (e.g., i in A minor and ii in G major).

Later on, the concept of *mehrdeutigkeit* was developed more extensively by the German music theorist Gottfried Weber (1779–1839), whose treatises *Versuch* discussed many aspects of

chord labelling ambiguity in the 19th century (Saslaw 1992, 93). In this section, I will use "Gottfried Weber and the concept of *Mehrdeutigkeit*" (Saslaw 1992), a dissertation that is largely based on the four volumes of the third edition of *Versuch*, as the primary source of *mehrdeutigkeit*.

4.1.1 Introduction of *mehrdeutigkeit*

In *Versuch*, Weber proposed a formal definition of *mehrdeutigkeit (multiple meaning)*: "The possibility of explaining an entity in more than one way" (Weber 1832, vol. I, 42, translated by Saslaw (1992, 94)). I consider it as the synonym of ambiguity, as defined in Glossary. Weber specified several general cases of *mehrdeutigkeit* in music:

- Every key of a piano has multiple meanings, since each one can be associated with several different pitch spellings (Saslaw 1992, 94).
- An interval can have multiple meanings if represented as the number of semitones from the lower note to the higher note. For example, an interval of six semitones can be a diminished fifth (C-G♭) or an augmented fourth (C-F♯) (Saslaw 1992, 109).
- The identification of voice crossing can also involve multiple meanings. Weber defined voice parts as the combination of upper, middle, and lower voices, where the first and last ones may also be considered as outer voices (Saslaw 1992, 112). How might one identify instances where when the parts meet or cross (Saslaw 1992, 112) within the same staff? One interpretation is to always consider the lowest note as part of the lower voice, and the highest note as part of the upper voice, eliminating the possibility of them crossing each other. The other interpretation is to consider voice-leading and decide when there is a voice crossing. However, the guidelines are not always clear, leading to a degree of ambiguity in identifying voice crossing.
- The interpretation of a composite melody can also involve multiple meanings. A composite melody refers to a single melodic line that can be perceived as a combination of several individual streams. The identification of each stream can lead to multiple interpretations (Saslaw 1992, 116). A simple melodic line with two possible interpretations is shown in Figure 4-1.

Figure 4-1: Illustration of the two possible interpretations of a melodic line, where the first one (left) contains two individual streams, and the second one (right) only contains one stream (Saslaw 1992, 117).

4.1.2 Mehrdeutigkeit in chord labelling

Weber used the diatonic tones of a key to define seven fundamental chord qualities, which are major, minor, and diminished triads; major, minor, half-diminished, and dominant 7th chords (Saslaw 1992, 123) shown in Figure 4-2. In *Versuch*, *mehrdeutigkeit,* (ambiguity) in chord labelling also concerns two more aspects: The identification of (non-)chord tones (Section 4.1.2.1) and mapping chord tones to chord labels, considering both omission (Section 4.1.2.2) and pitch spelling (Section 4.1.2.3).

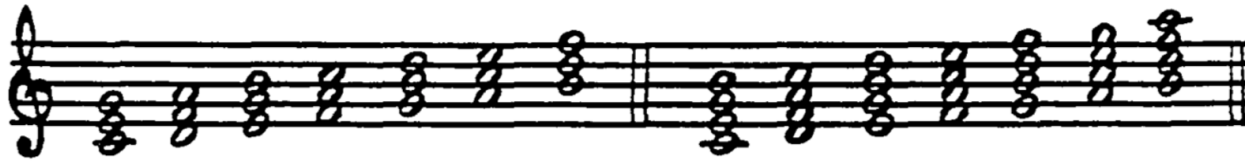


Figure 4-2: Triads (left) and tetrads (right) built on the diatonic tones of a C major scale (Saslaw 1992, 124).

4.1.2.1 Ambiguity in non-chord tone identification

According to Weber, any note that does not belong to the seven fundamental chord qualities (see Section 4.1.2) should be considered foreign to harmony (Saslaw 1992, 128), and can be referred to as non-chord tones (NCTs). It is possible that even some notes that belong to these chord qualities can potentially be NCTs, which can often be understood as *transition tones* (Saslaw 1992, 239); such notes are usually departed or approached by step. There are mainly five kinds of transition tones mentioned in *Versuch*: Appoggiaturas, accented appoggiaturas, passing tones, neighbour tones, and suspensions (Saslaw 1992, 240). Identifying which transition tones are NCTs can be particularly ambiguous (Saslaw 1992, 241). For example, the pitch class collection C-E-G with a passing B (shown in Figure 4-3) can be interpreted in two ways: One is to incorporate B as

a chord tone, making it a C major seventh chord, and the second is to consider B as a non-chord tone, resulting in a C major triad (Saslaw 1992, 245).

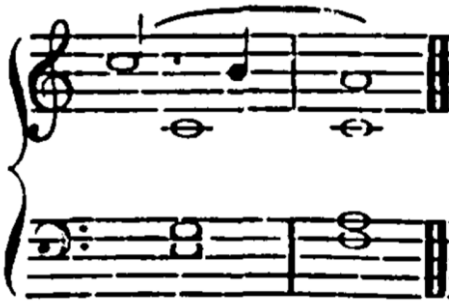


Figure 4-3: Illustration of a passing tone (B) that can either be interpreted as either a chord tone or a non-chord tone (Saslaw 1992, 245).

A more complicated case occurs when transition tones from different voices appear at the same time, causing further ambiguity as to whether these tones are NCTs or not. Three examples are shown in Figure 4-4, where the second simultaneity in each example contains a set of transition tones that can either be interpreted as NCTs or as G major triads (Saslaw 1992, 249).

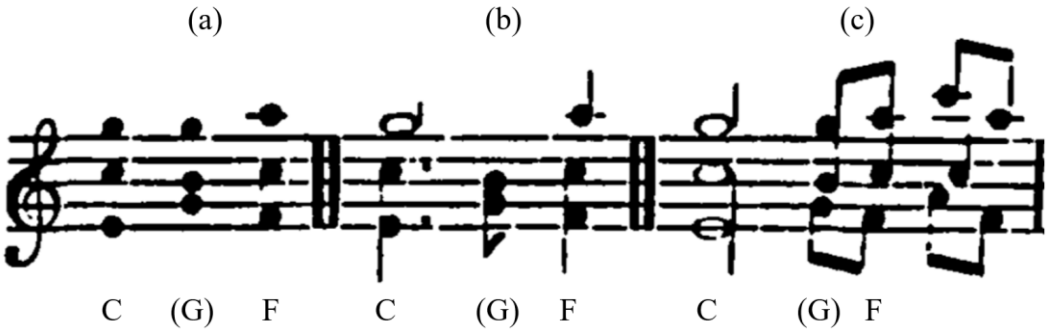


Figure 4-4: Three examples where transition tones occur simultaneously, which can all be identified as NCTs or form a passing G major triad. Image modified from Saslaw (1992, 249).

To help address the many alternative possible interpretations of NCTs, Weber recommended a few general rules (Saslaw 1992, 247–248):

- The first rule is that, if the interpretation of harmony leads to many chord labels within a bar (e.g., the upper track of chord labels in Figure 4-5), and there are many transition tones, especially on unaccented beats, and particularly in a fast tempo, then Weber prefers them to be treated as NCTs, resulting in fewer chord labels (e.g., the lower track of chord labels in Figure 4-5).

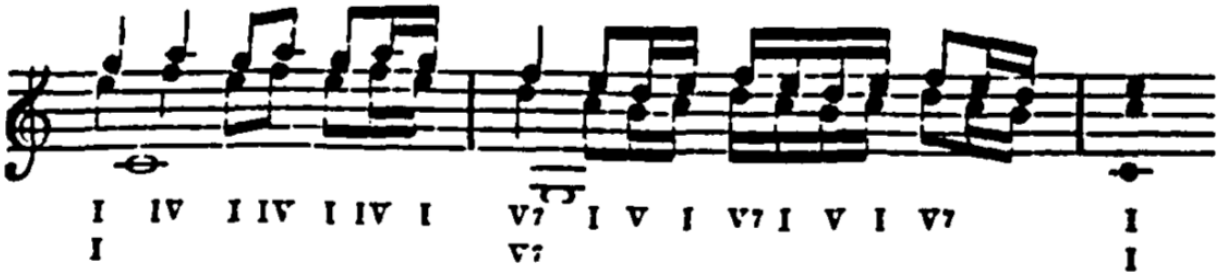


Figure 4-5: Two possible interpretations of chord labels for a musical passage with many transition tones, which can either be treated as chord tones, resulting in many chord labels (the upper track), or as NCTs, resulting in fewer chord labels (the lower track). Image from Saslaw (1992, 248).

- The second rule is that, if incorporating certain transition notes would cause a momentary digression to other keys, then the NCT interpretation is preferred. In examples (a) and (b) of Figure 4-6, the non-diatonic pitches in m. 1 (F♯ in (a), F♯ and D♯ in (b)) from Haydn's Symphony No. 103, can be regarded as passing NCTs, rather than forming new, non-diatonic chords.

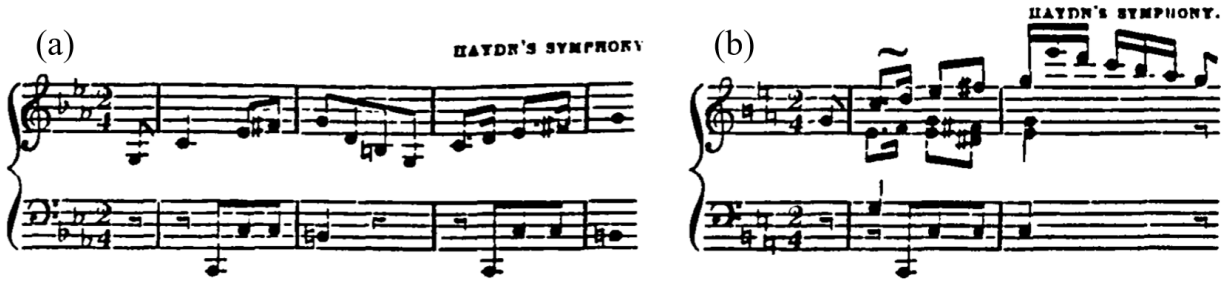


Figure 4-6: Two example passages of Haydn's Symphony No. 103. *Image modified from Saslaw (1992, 248).*

4.1.2.2 Ambiguity when chord tones are omitted

Another source of ambiguity comes from the absence of chord tones; as Weber stated: “It is easy to see that a chord consisting of so few members, or generally, a harmony in which one or several chord-tones are omitted, always acquires multiple meanings.” (Weber 1832, vol. I, 234, translated by Saslaw (1992, 127)). For example, the dyad A-C can be interpreted as two different triads (Saslaw 1992): either an A minor chord omitting the 5th or an F major chord omitting the root. Similarly, the dyad C-G can be interpreted as either an incomplete C major or C minor chord, depending on whether analysts identify E or E♭ as the implied chord.

4.1.2.3 Ambiguity associated with pitch spelling

Pitch-spelling can also involve ambiguity, resulting in different chord labels. Fully diminished chords are such a case: They contain a combination of minor third and augmented second intervals, and these two intervals contain the same number of semitones on the keyboard (Saslaw 1992, 129). As Weber noted: “In other words, according to our tempered tone-system, all the tones of such a chord are equally distant from each other” (Weber 1832, vol. I, 243, translated by Saslaw (1992), 129)). As a result, these chords can have four different labels depending on how the pitches are spelled (Saslaw 1992, 130). For example, F♯o7 (F♯–A–C–E♭) is enharmonically equivalent to three other inverted diminished-seventh chords with different spellings and chord roots: (1) Ao7 (G♭–A–C–E♭), (2) Co7 (G♭–B♭♭–C–E♭), and E♭o7 (G♭–B♭♭–D♭♭–E♭), as shown in Figure 4-7. In symbolic music, ambiguity of this kind can happen when chord labelling is conducted on a MIDI input (no pitch spelling information). However, this kind of ambiguity can be resolved if chord labelling is conducted on a musical score, where the spelling of each pitch class is explicitly specified.

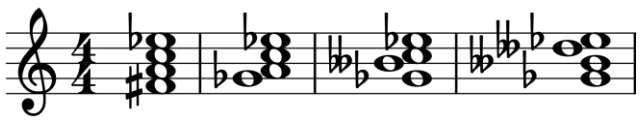


Figure 4-7: The same four enharmonically equivalent pitch classes spelled as four different diminished seventh chords: F♯o7, Ao7, Co7, and E♭o7.

4.1.3 Summary

In this section, I introduced a brief overview of chord labelling ambiguity in music theory, and much of the discussion is based on the discussions of Weber's treatises *Versuch* from Saslaw (1992), which offers great insights on this topic. The discussion in Section 4.1.2 in particular served as a fundamental source for the methodologies on how chord labelling ambiguity can be interpreted, as proposed in Section 1.4. Next, I will expand this discussion by introducing perspectives on chord labelling ambiguity from music information retrieval.

4.2 Chord labelling ambiguity in music information retrieval

A primary goal of music information retrieval (MIR) research is to access information from or about music (in audio or symbolic formats, or from other information sources) automatically. This can include information relating to musical aspects such as musical forms, keys, chord labels, etc. Many automated approaches are based on supervised machine learning, and require a set of manually annotated data for training and evaluation. The sections below highlight certain insights from the MIR community relating to chord labelling ambiguity.

4.2.1 Inter-rater variability in chord labelling

The chord labels prepared by different experts can be significantly different. These differences can be referred to as *inter-rater variabilities*,[31] which have been investigated in MIR in the past in audio chord labelling:[32] De Clercq and Temperley (2011) reported a disagreement rate of 7.6% among two annotators when transcribing chord roots (relative to the key, such as scale degrees) in 100 rock songs; Ni et al. (2013) found that there was about 10% of major or minor triads (e.g., "G" triad) six reference annotations disagreed when transcribing 20 songs by The Beatles and Queen. Koops et al. (2019) reported a much higher rate of disagreement among annotators when transcribing chord labels for the songs from Billboard: 27% for major and minor triads and 46% for the more complex chords. As a result, these statistics suggest a "subjectivity

[31] To my best knowledge, comparing chord labels prepared by the same expert at different times, i.e., the study of *intra-rater variabilities*, has not been attempted in chord labelling.
[32] To my best knowledge, there is no such study in symbolic music, where inter-rater variabilities were studied in a quantifiable way.

ceiling" in harmony, meaning that, if a dataset is labelled by several experts whose internal harmonic models are different from each other, it is hard to train an automated model that agrees more than what experts agree with each other, resulting in an upper bound for evaluations in the computational research of harmony (Koops 2019, 30).

Although certain chord labels prepared by different experts can be different, these chord labels may still be related or similar. Take the chord labels in Figure 4-8 as an example:

- G and GM7 at m. 2.1.5: They both share the same chord root and the same harmonic function (tonic).
- G and Em7 at m. 1.1.5: Although they do not share the same chord root, the chord tones of G are the subset of Em7, and they share the same harmonic function (tonic).
- Am and F♯o at m. 1.4.5: They are two common chord tones: A and C, and they also share the same harmonic function (dominant).

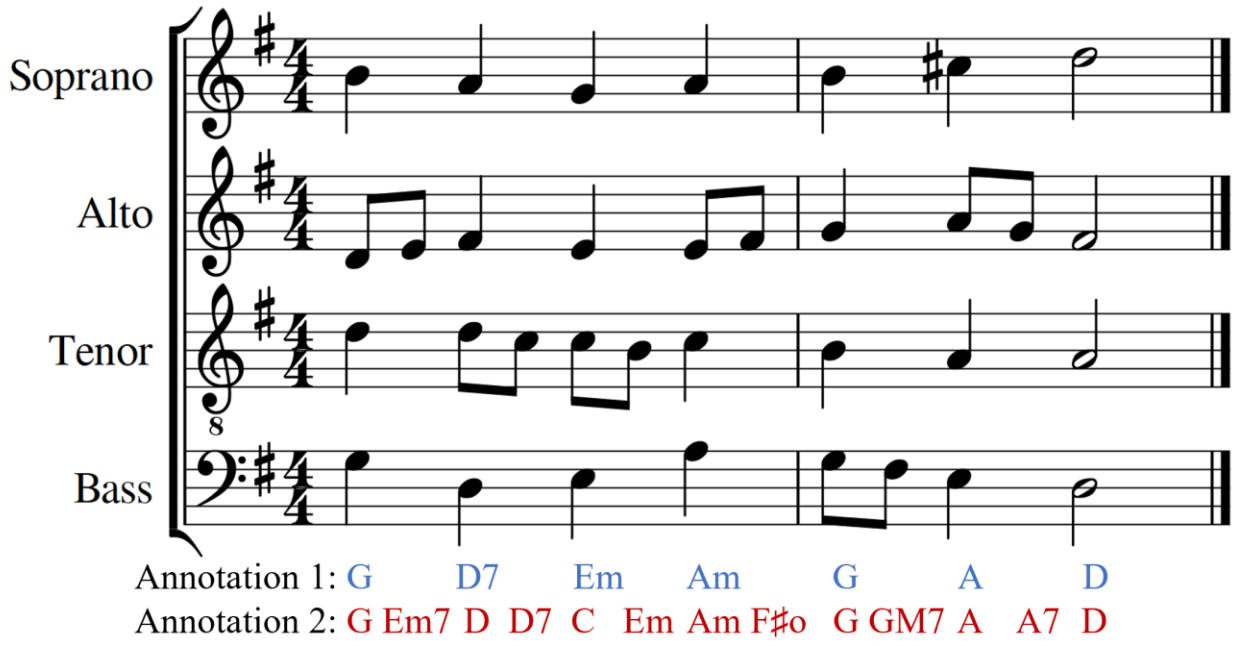


Figure 4-8: A musical passage that can be annotated with chord labels in two different ways. Image modified from Ju et al. (2019, 862).

Although unweighted classification accuracy is often used as an evaluation metric in automatic chord labelling (see Section 3.4.7), this metric only accounts for exact matches, and does not consider how different chords are related. To better demonstrate the relationship between two different chords, metrics of chord distance and chord similarity can be considered: They will respectively be introduced in Section 4.2.2 and Section 4.2.3.

4.2.2 Chord distance metrics

In this section, I will introduce two chord distance metrics. One is proposed by Chuan and Chew (2008), which is based on the chord substitution theory introduced by the German theorist Bernard Riemann (1826–1866). According to Riemann, there are three basic harmonic substitutions (see Section 3.1.2.3 for details): *Variante* (V)*, Parallele* (P)*, and Leittonsweschel* (L). As shown in Figure 4-9, chords can transform into one another using one or more substitution rules. For example, "i" can be transformed into "I" using a single "V" transformation, "v" into "vii" using "V" and "L" transformations, and "i" into "III" using "V", "L", and "V" transformations. According to Chuan and Chew (2008), the number of transformations is defined as the distance between two chords, which is respectively one, two, and three for the examples above. This metric is simple and straightforward, but it only works for major or minor triads, and the distance cannot be measured for more complex chords.

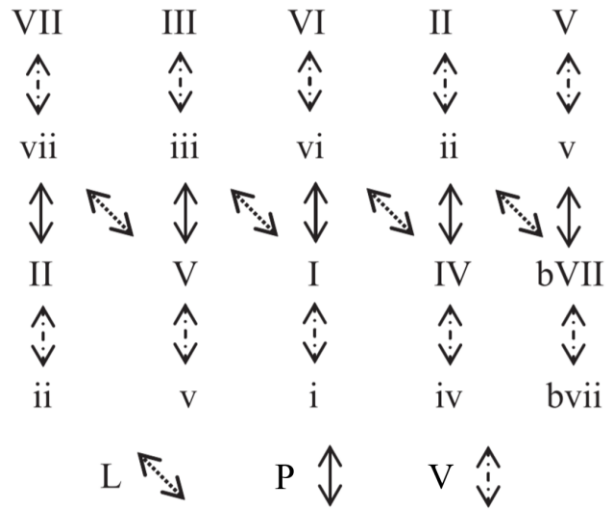


Figure 4-9: Illustration of chords that can be transformed into one another using three basic substitution rules: Leittonsweschel (L), Parallele (P), and Variante (V) introduced in Section 3.1.2.3. Image modified from Chuan and Chew (2008, 58).

A second metric of chord distance is proposed by Haas, Wiering, and Veltkamp (2013), which is based on Tonal Pitch Space (TPS) (Lerdahl 2004), a model built on insights from the *Generative Theory of Tonal Music* (Lerdahl and Jackendoff 1983), and is designed to reflect music-theoretical and music-cognitive intuitions about tonal organization. This metric represents chords in a given key with the *basic space,* and is essentially a scoring function that is intended to consider the perceptual importance of chord tones. As shown in Figure 4-10, there are five levels in the basic space, with chords expressed in three levels: (a) Root level; (b) fifths level, and (c) chord tone level; and keys expressed in two levels: (d) Diatonic level and (e) chromatic level (De Haas, Wiering, and Veltkamp 2013).

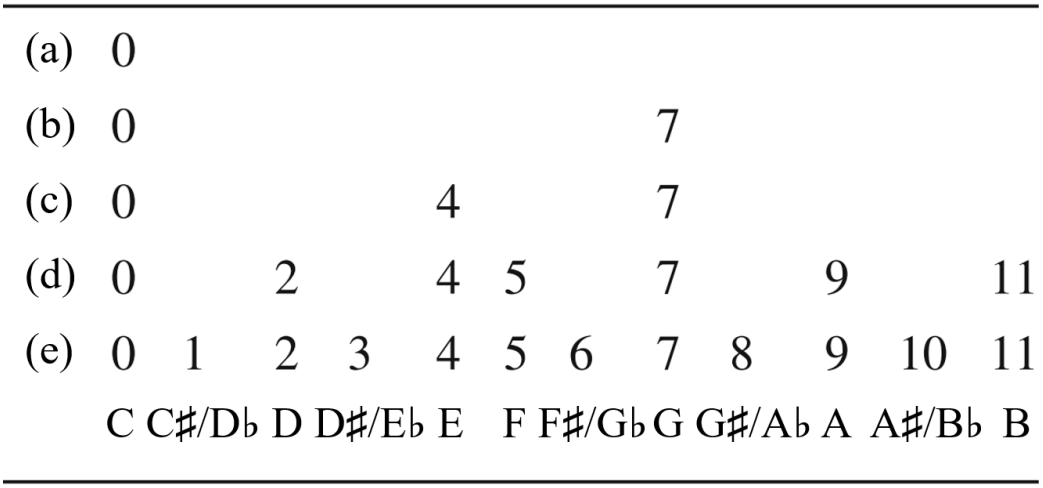


Figure 4-10: Illustration of the basic space representing a C major triad in the key of C major, where the enharmonic pitch classes are represented as numbers 0 to 11 in five different levels. (a) root level; (b) fifths level; (c) chord tone level; (d) diatonic level; (e) chromatic level. Image modified from De Haas, Wiering, and Veltkamp (2013, 193).

Let c_1 and c_2 be the two target chord labels, and C_d be the chord distance function given the two chords (c_1, c_2) in a given key (k), which can be measured as:

$$C_d(c_1, c_2, k) = m + n,$$

$C_d(c_1, c_2, k)$ is usually a value between 0 and 13. m indicates the minimal number of steps between the roots of two chords in the line-of-fifths (see Figure 4-11), and n indicates the number of pitch classes that are *distinct* in the levels (a)–(d) when comparing the basic space of c_2 against that of

c_1. A pitch class will be considered distinct if it is found in one basic space but not in the other (De Haas, Wiering, and Veltkamp 2013).

Figure 4-11: The line of fifths (Temperley and Sleator 1999, 16).

(a)					4							
(b)					4							11
(c)					4			7				11
(d)	0		2		4	5		7		9		11
(e)	0	1	2	3	4	5	6	7	8	9	10	11
	C	C♯/D♭	D	D♯/E♭	E	F	F♯/G♭	G	G♯/A♭	A	A♯/B♭	B
(a)			2									
(b)			2							9		
(c)			2			5				9		
(d)		1	2		4	5	6	7		9		11
(e)	0	1	2	3	4	5	6	7	8	9	10	11
	C	C♯/D♭	D	D♯/E♭	E	F	F♯/G♭	G	G♯/A♭	A	A♯/B♭	B

Figure 4-12: The basic space transformation from a G chord to an Em chord, in the key of C major (top) and the basic space transformation from a D chord to a Dm chord, in the key of the D major (bottom). The distinct pitch classes are underlined. Image modified from De Haas, Wiering, and Veltkamp (2013, 194).

Take Figure 4-12 as an example, the upper example is the Em chord's basic space in the C major key, which demonstrates the calculation of the distance between the G chord and the Em chord. At the root level (a), G and Em do not share the same chord root, resulting in a distinct pitch class; at the fifths level (b), G and Em do not share the chord root nor the fifth, resulting in two distinct pitch classes; at the chord tone level (c), there is one unique chord tone in each of the two chords ("D" in the G chord, and "E" in the Em chord), resulting in one distinct pitch class. Since the chord tones of G can all be found in the key of C major, there is no distinct pitch class at level

(d). Therefore, the numbers of distinct pitch classes at these five levels are 1, 2, 1, 0, respectively, and the total number of pitch classes n is 4. Since there are three steps between chord roots G and E (on Figure 4-11), the minimal number of steps in the line-of-fifths $m = 3$. Therefore,

$$C_d(G, Em, C\ major) = 4 + 3 = 7.$$

Similarly, the distance between the D and Dm chords in the key of D major is:

$$C_d(D, Dm, D\ major) = 2 + 0 = 2.$$

4.2.3 Chord similarity metrics

The relationship between two chord labels can also be measured as chord similarity. To discuss this, let us adopt the following notation: C_s is a chord similarity function, and P_c represent all the pitch classes in a given chord.

The first of the two metrics introduced in this section is the Harte Distance Metric (Harte 2010), which takes the size of the intersection of the pitch-class sets of both chords and divides it by the size of the union of the pitch-class sets of both chords, as follows (Freedman 2015, 19):

$$C_s(c_1, c_2) = \frac{|P_c(c_1) \cap P_c(c_2)|}{|P_c(c_1) \cup P_c(c_2)|}.$$

For example, the score when comparing the minor and major chords with the same chord root (e.g., $\mathrm{C_s}(D, Dm)$) and a major chord with its relative minor chord (e.g., $\mathrm{C_s}(D,\ Bm)$) will both result in the same value of 0.5; a major triad comparing to a major seventh chord with the same chord root (e.g., $\mathrm{C_s}(E, Emaj7)$) will result in a score of 0.75. Comparing chords with the same pitch sets, even though the chord root or lowest note is different (e.g., $\mathrm{C_s}(D6, Bm7)$ and $\mathrm{C_s}(D7, D7/\mathrm{C})$, will result in a score of 1. We can see that despite being a simple chord similarity metric, Harte Distance Metric only focuses on the pitch classes of chords and discards some meaningful distinctions of chords, such as chord root and lowest note (Freedman 2015, 19).

The second chord similarity metric $C_s(c_1, c_2, k)$ is derived from the chord distance metric $C_d(c_1, c_2, k)$ introduced in Section 4.2.2. To make it as a fraction between 0 and 1, similar to the Harte Distance Metric, Freedman (2015, 21) normalized the value of $C_d(c_1, c_2, k)$, in which 1 indicates being identical:

$$C_s(c_1, c_2, k) = \frac{13 - C_d(c_1, c_2, k)}{13}.$$

4.2.4 Summary

This section discussed how chord labelling ambiguity has been addressed in music information retrieval, including studies on inter-rater variability between different experts, as well as metrics based on chord distance and chord similarity that reflected the relationship between different chords in a quantifiable way. Some of the chord distance metrics have been used to compare chords or chord sequences before. For example, Chuan and Chew (2008) used their chord distance metric in the task of melody harmonization: They compared the predicted chord labels (based on a given melody) against ground truth chord labels, and De Haas, Wiering, and Veltkamp (2013) modified their chord distance metric to compare two chord sequences. Although these metrics have not been used in automatic chord labelling, they could serve as invaluable resources and a starting point for addressing the ambiguity in automatic chord labelling. In Section 4.3, I will discuss an overview of methodologies I propose to address ambiguity and chord labelling, based on a collection of insights from Chapters 2, 3, and the rest of 4.

4.3 Addressing ambiguity in automatic chord labelling: Proposed methodologies

Up until this point, the existing literature introduced above is mainly about chord labelling ambiguity: How it originated from a historical perspective (Section 3.3), specific treatises or papers discussing it in music theory and MIR (Section 4.1 and Section 4.2, respectively), and how ambiguity can be addressed in machine learning (Chapter 2). The discussion below brings together nuances from the different disciplines, and proposes methodologies for addressing ambiguity in automatic chord labelling.

4.3.1 Discussion

Due to ambiguity, chord labels provided by different experts or even the same expert can vary significantly, resulting in a harmonic “subjectivity ceiling” as an upper bound for evaluations

in the computational research of harmony (Koops 2019, 30) (see Section 4.2.1). There are three possible approaches to this problem:

1. One is to use chord distance and/or chord similarity as evaluation metrics, so that when the generated chord label is not identical with the ground truth, they can still be related, as discussed in Section 4.2.1, and a weighted score can reflect this connection.
2. As discussed in Flexer and Lallai (2019), another solution is to reduce these variabilities and ensure the consistencies of these chord label annotations as a single reference ground truth. This can be achieved by, for example, using a more controlled group of annotators, carefully curating materials that are less ambiguous to analyze (Flexer and Lallai 2019).
3. Alternatively, one can also embrace these variabilities by incorporating parallel tracks of chord labels as multiple reference ground truths, where each reflects a different chord labelling strategy.

4.3.2 Proposed methodology

In this dissertation, I will adopt the second and the third approaches to address ambiguity in automatic chord labelling, using Bach chorales as a case study. I chose this repertoire due to its key role in modern music pedagogy and its general historical importance. The first approach has not been explored for the time being, but it will be valuable to implement in future research.

The first step is to use a single reference ground truth and reduce the annotation variabilities. As discussed in Section 3.3 and Section 4.1, these variabilities may not be a result of random choices, but rather could be systematic preferences, originating from different analytical perspectives in music theory. To ensure the consistency of chord label annotations, I begin by introducing a rule-based algorithm that generates preliminary chord labels according to a specific analytical perspective specified by users. These chord labels are later checked by an expert who provides corrections when necessary. Compared to generating annotations from scratch, the amount of required work for the expert is significantly reduced, and the resulting annotations combine the consistency of the rule-based model with the nuance of manual corrections. These annotations are then used to train automatic chord labellers. The details of this approach will be introduced in Chapter 5.

Then, I will focus on obtaining parallel tracks of reference chord labels; a series of four rule-based algorithms is proposed to automatically generate chord labels for Bach chorales based on both figured bass annotations and the musical surface. The resulting chord labels will be presented as the new Bach Chorales Multiple Chord Labels (BCMCL) dataset in Section 6.5.

Finally, I will use the BCMCL dataset to explore multi-label learning and label distribution learning for automatic chord labelling, an under-studied but essential aspect of music information retrieval research. The parallel chord labels from BCMCL can also serve as a basis for exploring alternate evaluations for automatic chord labellers in general. The automatic chord labellers resulting from this research can generate either single chord labels according to a specific analytical strategy, or can generate multiple labels for a single chord based on a variety of different analytical perspectives. The details of this approach will be introduced in Chapter 7.

4.4 Conclusion

In this chapter, I discussed the existing literature on chord labelling ambiguity in music theory and music information retrieval (MIR), including a few chord distance and chord similarity metrics. In Chapters 5, 6, and 7, I will provide details on approaches to addressing chord labelling ambiguity in the domain of automatic chord labelling.

Chapter 5 Building automatic chord labellers using a single ground truth

The research introduced in this chapter will attempt to answer the second research question proposed in Section 1.6:

- If single-label learning is used for automatic chord labelling, what are the ways to generate single labels for chords?

As discussed in Section 3.3, ambiguities in chord labelling are not only a result of arbitrary choices but also systematic preferences, which originate from different analytical perspectives in music theory. One way of generating single labels for chords is to precisely define a specific analytical perspective, so that the resulting annotations can have a definitive label for each chord. In the work described in this chapter, a rule-based algorithm is proposed, which can generate variant chord labelling analyses based on different analytical perspectives (Section 5.1). In Section 5.2, I will first precisely define an analytical strategy, so that the rule-based algorithm will generate single labels for chords. These preliminary annotations will later be modified by one music theory expert with more nuance. The resulting modified annotations combine the consistency of the rule-based model with the nuance of manual corrections and can be used to train and evaluate automatic chord labellers using single-label learning (Section 5.2). Finally, the conclusions of this work will be given in Section 5.3.

5.1 The rule-based algorithm

The work introduced in this section has been published in the 19th International Society for Music Information Retrieval Conference (Condit-Schultz, Ju, and Fujinaga 2018). Therefore, most of the text in the following sub-sections (within Section 5.1) is used verbatim without using quotation marks from this paper. In Section 5.1.1, I will introduce the method of generating single labels for chords by proposing a specific analytical strategy. This strategy can be defined by choosing a consistent answer to each step of the chord labelling process discussed in Section 5.1.1.1, Section 5.1.1.2, and Section 5.1.1.3. In Section 5.1.3, a rule-based algorithm that can

generate variant chord labelling analyses based on different analytical perspectives is proposed, which is applied to the chorale music we compile in Section 5.1.2. The co-author Nathaniel Condit-Schultz and I proposed the idea of building a rule-based algorithm, and the design and the implementation of the algorithm introduced in Section 5.1.2 and 5.1.3 are primarily by Nathaniel Condit-Schultz.

5.1.1 Defining an analytical strategy

One way to generate single labels for chords is to precisely define an analytical strategy. This strategy can be defined by choosing a consistent answer to each step of the chord labelling process, which will be introduced in Section 5.1.1.1, Section 5.1.1.2, and Section 5.1.1.3. Finally, a musical example will be given in Section 5.1.1.4 to illustrate this process.

5.1.1.1 Defining chord qualities

In the process of chord labelling, analysts may choose a specific set of chord qualities, depending on their preferences, music training, and the type of music to analyze. Ambiguity will arise if two analysts have different chord qualities considered. For example, if one analyst considers 9^{th} chords, and the other analyst only considers triads and 7^{th} chords, their analyses will differ when ninths are involved in the music. We can unify this process by proposing a definitive set of chord qualities for chord labelling, and in the common practice period, for example, we can define that the chord qualities that should be considered to be: Major, minor, diminished, and augmented triads; major, minor, dominant, half-diminished, and diminished 7^{th} chords.

5.1.1.2 Identifying non-chord tones

Once the allowable chord qualities are decided, one can proceed with chord labelling by identifying non-chord tones and decide which notes of the musical surface do not comprise a chord. However, as discussed in Section 1.4, Section 3.3, and Section 4.1.2.1, the guidelines of how to identify non-chord tones are generally not clear, and there could be various possible interpretations, resulting in chord labelling ambiguity. Here, we will discuss the process of non-chord tone identification in detail and outline its ambiguity.

Given the allowable chord qualities, for example, if 9^{th} chords are not considered in chord labelling, there is no ambiguity on whether ninths should be non-chord tone or not—it should always be considered as non-chord tones. Likewise, if 7^{th} chords are not considered in chord labelling, all the sevenths must be considered as non-chord tones.

In the common practice period, triads and 7th chords are usually considered as candidate chord qualities, but this does not mean that any simultaneity that comprises any of these chord qualities should always be identified as such. Some of these notes, from the contrapuntal point of view, may exhibit transition motions (see Glossary) that belong to the following categories: Passing tone, neighbour tone, suspension/retardation, appoggiatura, escape tone, pedal tone, and double passing tone. In this case, these notes could be considered as non-chord tones that decorate or resolve to chord tones, or still be considered as chord tones. This flexibility will make the process of non-chord tone identification ambiguous, leading to the main source of ambiguity in chord labelling.

Analysts often employ different rules and preferences within this analytical flexibility, where some may provide analyses with more non-chord tones, while others may prefer analyses with fewer non-chord tones. As discussed in Section 3.3.4, three prominent music theorists: Rameau, Weber, and Schenker had different approaches to non-chord tone identification: Rameau's theory of supposition incorporated more notes as chord tones, compared to Weber who used voice-leading to identify notes in transitional motions as non-chord tones. Schenker, on the other hand, proposed multiple layers of chord labelling, each with a different balance between chord tones and non-chord tones, reflecting different analytical strategies.

5.1.1.3 Mapping chord tones into chord labels

Once non-chord tones are identified and removed from the musical surface, the final step of chord labelling is to map the chord tones into chord labels. Often there is no ambiguity involved in this step, since each slice will be labelled with a distinct chord label, but sometimes ambiguity will arise if the sonority of one slice is a subset of the other. For example, if the sonority of one slice is {C, E, G} and the following slice is{C, E, G, B}, these two slices can either be labelled with one chord label "CM7" or could respectively be labelled as "C" and "CM7". The former

option can be seen as preferring fewer chord labels and the latter can be seen as preferring more chord labels.

5.1.1.4 A musical example

Figure 5-1 illustrates the possible chord labelling analyses using a concrete musical example.[33] In this example, triads and 7th chords are defined as the candidate chord qualities. Therefore, the three notes coloured red must be interpreted as non-chord tones according to Section 5.1.1.1, since they are the ninths and cannot be incorporated as part of the chord. Blue notes exhibit transitional motions and can be classified as part of triads or 7th chords. According to Section 5.1.1.2, they can either be considered as non-chord tones or chord tones. This process is not arbitrary, and analysts often employ systematic rules to distinguish them.

The passing tones in Slices 2 and 8 are especially illustrative. If the passing tone in Slice 2 is considered a chord tone, Slices 1 and 2 form the harmonies I→vi^6, both tonic function chords. If the passing tone in Slice 8 is interpreted as a chord tone, the progression ii→vii^{o6} results in a transition between two different tonal functions (subdominant and dominant). Given these functional differences, many analysts would mark Slices 1 and 2 as a single I chord but Slices 7 and 8 as ii→vii^{o6}. This is especially true since transitions from ii→I (slice 9) are considered abnormal, while transitions from vii^{o6}→I are normative. Several slices illustrate the ambiguity regarding 7th chords: Passing tones in slices 6, 18, 20, 22 and 24 might each be either interpreted as chordal 7ths (chord tones), or non-chord tones. For instance, the G in Slice 11 can be seen as the 7th of a ii^6 chord, or as a suspension.

Using Figure 5-1 as an illustration, it seems that the main fundamental source of ambiguity in chord labelling is due to experts' divergent different analytical strategies, resulting in multiple alternate analyses. As discussed in Sections 5.1.1.1, 5.1.1.2, and 5.1.1.3, single chord labelling analyses can be produced once the definitive and consistent strategies for each of these three chord labelling aspects are given. To achieve this goal, a rule-based algorithm that computes all possible analyses which satisfy only basic, uncontroversial constraints (defined in Section 5.1.3) is

[33] This example and the corresponding discussions are contributed by the co-author Nathaniel Condit-Schultz of our paper (Condit-Schultz, Ju, and Fujinaga 2018).

developed. These myriad interpretations can later be filtered to extract specific analyses. For instance, users can specify whether to permit 7th chords (allowing 7ths to be chord tones) and the type of non-chord tones (see the specific categories in Glossary) as filters to generate the corresponding analyses. A detailed discussion on how the algorithm will generate various analyses, and how they can be filtered to fewer analyses will be given in Section 5.1.3.4.2 and illustrated in Figure 5-3.

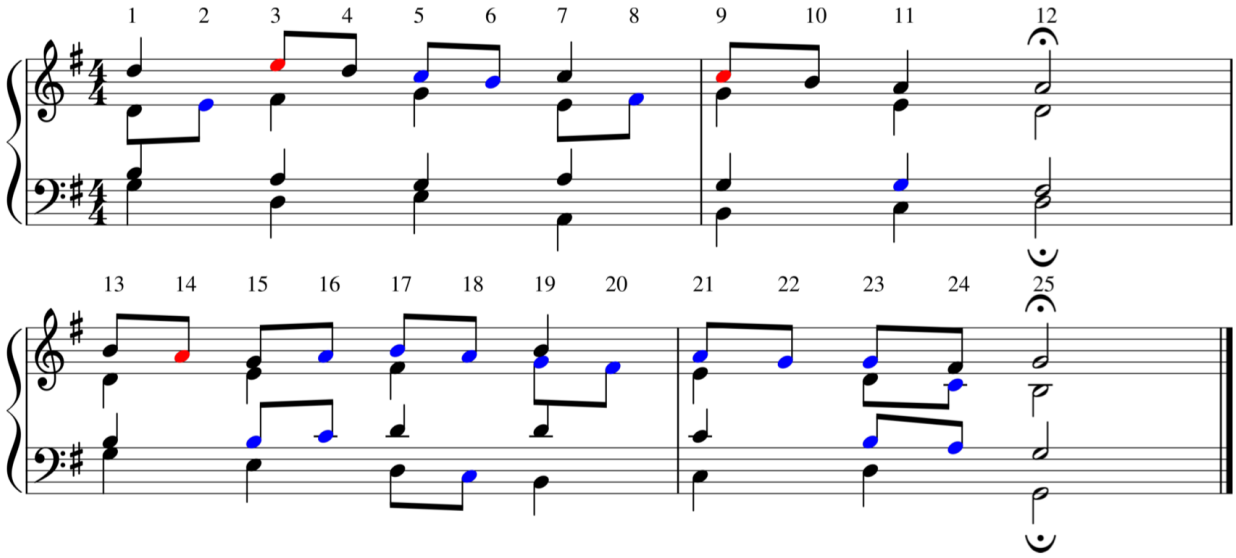


Figure 5-1: Illustration of how chord labelling ambiguity can be involved in a contrived example of a four-part counterpoint, composed by Nathaniel Condit-Schultz. Note onset slices are numbered above the staff. Notes coloured red indicate non-chord tones. Notes coloured blue exhibit transitional motions, including passing tones (Slices 2, 5, 8, 16, 18, 20, 21, 23, 24), neighbour tones (Slices 6, 17, 18, 19), suspensions (Slices 11, 23), retardation (Slice 15), and anticipation (Slice 22), which can be considered either as chord tones or non-chord tones. However, some of these interpretations are mutually exclusive, as a non-chord tone must resolve to a chord tone and usually cannot be followed by another non-chord tone. For instance, if the C in Slice 5 is considered a passing tone, then the B in Slice 6 must be a chord tone, which resolves the passing tone.

5.1.2 Dataset

The rule-based algorithm is applied to Baroque four-voice chorale music, namely the 371 chorales by Johann Sebastian Bach (1685–1750) and the 200 chorales by Michael Praetorius (1571–1621). Symbolic encodings of the Bach chorales were mostly gathered from the KernScores repository (kern.ccarh.org), which is maintained by Stanford's Center for Computer Assisted Research in the Humanities. It contains 370 four-part chorales, and we (Condit-Schultz, Ju, and

Fujinaga 2018) manually encoded one extra five-part chorale and added it to the dataset for the purpose of this study. All 371 chorales are encoded in the humdrum **kern representation (www.humdrum.org) (Huron 1999). Symbolic encodings of 200 chorales by Praetorius were recently digitized from the 1928 Friedrich Blume edition (Blume 1928–1940), where scores were initially scanned and interpreted by Vi-An Tran using optical music recognition software and then corrected by Carlotta Murtano, both studied at McGill University. The Praetorius data includes 197 four-voice chorales and three five-voice chorales. In total, the dataset includes 571 chorales encoded in **kern format, consisting of 129,568 notes (plus 898 rests), which form 42,895 note onset slices. The dataset includes 571 chorales composed by Bach and Praetorius and is available at: https://github.com/DDMAL/Flexible_harmonic_chorale_annotations. This dataset can serve several useful functions:

- Researchers can generate specific, consistent chord labelling analyses, conforming to whatever analytical strategies they prefer, for all music in the corpora. These varied analyses present a potentially valuable comparative resource, which may also be of musicological and music-theoretical interest. They can be used like any other chord label annotations—i.e., to study harmonic progression and tonality in general.[34]
- The dataset includes a set of late-modal (Praetorius) and early tonal (Bach) music, which are nonetheless largely similar in texture and style. This makes the dataset particularly useful for style comparison and analysis (Hedges and Rohrmeier 2011).

Chorale music is invaluable for teaching and studying harmony, as it features consistent and highly constrained contrapuntal textures. The 371 Bach chorales are mainstays of music theory pedagogy and have been the subject of much music information retrieval research (De Clercq 2015; Hadjeres, Pachet, and Nielsen 2017; Hedges and Rohrmeier 2011; Kröger et al. 2008; Quinn 2010; Quinn and Mavromatis 2011; Quinn and White 2013; Rohrmeier and Cross 2008; Woolhouse 2015). Several sets of expert chord label annotations for Bach's chorales have been published (Sapp 2007; Kröger et al. 2008; Cuthbert and Ariza 2010; Radicioni and Esposito 2010) with

[34] All the 371 Bach chorales with one possible set of chord label annotations can be found at: https://github.com/juyaolongpaul/harmonic_analysis/tree/master/genos-corpus/answer-sheets/bach-chorales/New_annotation/ISMIR2019, where each **kern file contains an extra spine of `**chordsymbol` output by the rule-based algorithm. These annotations are used in Section 5.2.2.3 for automatic chord labelling.

annotations digitally aligned with symbolic music data. Although the 200 Praetorius chorales share a similar texture with Bach chorales, they have received relatively little attention.

5.1.3 Methodology

The approach of this project is to calculate various possible chord labelling interpretations, and only to filter out specific interpretations at a later stage. The main goal is to establish "basic" constraints on harmonic interpretation. In true music theory form, these constraints can be formulated as the "rules" specified below. There are two types of rules to focus on: Harmonic rules and melodic rules, which are an amalgam of the rules explicitly, or implicitly, described in typical music theory textbooks (Kosta 2000; Laitz 2012). These rules are specialized (through some trial and error) for our chorale datasets, therefore they may not apply well to other repertoires.

5.1.3.1 Harmonic rules

Harmonic rules are specified as follows:

1. Every sonority slice can be associated with one and only one chord label.
2. Every chord label cannot sustain through a metric position stronger than where it starts—i.e., harmonic rhythm cannot be syncopated.
3. Only triads (major, minor, diminished, or augmented) and 7th chords (dominant, major, minor, half-diminished, or fully diminished) are considered legal chord qualities. However, incomplete chords may also appear in music. Complete chords are preferred, but cardinal-three subsets of seventh chords (Root-3rd-7th or Root-5th-7th), dyadic subsets of triads (i.e., consonant intervals), and even unisons/octaves are permitted. Here, we consider only triads and 7^{th} chords, so there is no ambiguity on which chord qualities should be considered, as discussed in Section 5.1.1.1.

Given these rules, we can then establish which notes do, or do not, belong to the local harmony. To be a non-chord tone, a note must satisfy all of the following melodic rules—any note that fails any of these rules must be a chord tone:

5.1.3.2 Melodic rules

1. The antecedent and consequent note of each non-chord tone must be chord tones, except for the special case of double passing (see definition in Glossary).
2. Non-chord tones cannot sustain across metric positions that are stronger than their own metric position. For example, a non-chord tone on a weak beat cannot sustain across the next down beat.
3. A note cannot start as a non-chord tone and then become a chord tone (though the opposite is possible, in suspension).
4. Finally, all non-chord tones must exhibit transitional motions, which are either departed or approached by step, including the ones (passing tone, neighbour tone, suspension/retardation, appoggiatura, escape tone, pedal tone, and double passing tone) defined in Glossary.

5.1.3.3 Data parsing

The **kern encodings of the dataset were parsed using the Humdrum Toolkit, before being loaded into R (Team 2013) for additional parsing. The analysis workflow was also conducted in R. To make the analyses useful as comparisons across the two composers, the same parsing and analysis workflow are applied to each.

In addition to pitch and rhythm data, the Bach chorale data contains fermatas as phrase boundaries. A phrase ending in a Bach chorale was identified whenever all four voices reach a fermata.[35] The Praetorius chorale data contains phrasing information, encoded as rests in all voices, and both datasets contain metric information.

5.1.3.4 Workflow

Our process has a two-stage workflow. The first stage (Section 5.1.3.4.1) is to divide the music exhaustively into contiguous groups of successive note onset slices as "contextual windows",

[35] Several chorales had notational inconsistencies, wherein fermatas were not encoded on the inner voices. These inconsistencies were fixed manually.

shown in Figure 5-2. The second stage (Section 5.1.3.4.2) applies an 8-step algorithm that automatically generates analyses (chord letters) to the slices in each window.

5.1.3.4.1 Stage 1

Many sonority slices can be analyzed in isolation. However, many more slices need context to be analyzed. Our approach is to parse the music into a single set of contiguous (non-overlapping) windows, identified using a simple, rule-based heuristic. A new contextual window begins anytime when any of these conditions are met:

- All voices attack on a strong beat.
- All voices attack and one or more voices did not attack in the previous onset slice.
- In an offbeat slice, more than two voices attack, and one or more voices sustain into/past the next beat.
- A phrase boundary is encountered, as indicated by either fermatas (Bach chorales) or rests in all voices (Praetorius chorales) as discussed in Section 5.1.3.3.

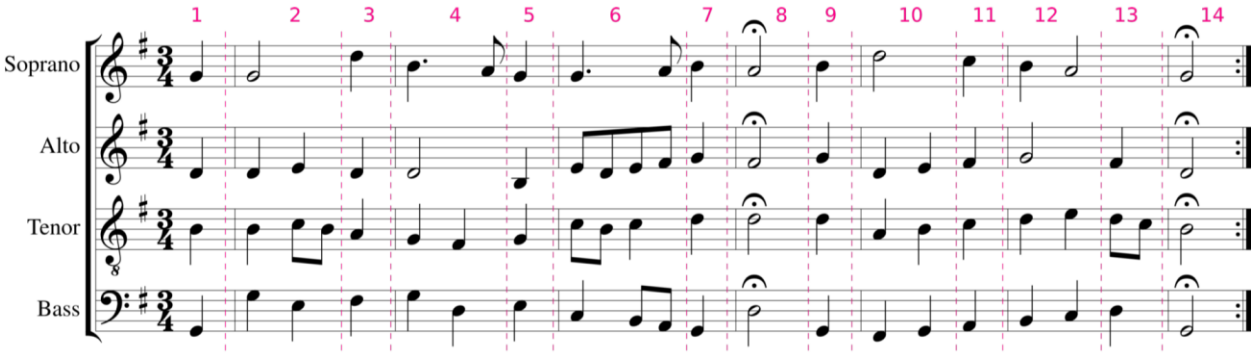


Figure 5-2: Illustration of contextual windows in BWV 269, Aus meines Herzens Grunde. *Slices between dashed red lines are analyzed as one window.*

Figure 5-2 illustrates the contextual windows derived by these heuristics in the first fourteen measures of music. The goal of this step is to determine a preliminary version of harmonic rhythm where each window will be associated with at least one chord label. The next stage will identify the number and the kinds of chord labels for each window.

5.1.3.4.2 Stage 2

Once contextual windows are identified, the following algorithm (8 steps) is applied to the slices in each window, where chord qualities can be specified among all the legal ones we considered in Harmonic Rule 3 (including major, minor, diminished, or augmented triads and dominant, major, minor, half-diminished, or fully-diminished 7th chords). The output is a range of chord label interpretations illustrated in Figure 5-3:

1. Identify all ways in which the window can be divided exhaustively into sub-segments. Take Figure 5-3 as an example. Since there are four onset slices in this window, there are eight possible segmentations: "1…", "12..", "1.2.", "1..2", "12.3", "1.23", "123.", and "1234", where the digit indicates the ID of a sub-segment and the dot indicates that a previous subsegment sustains over the current onset slice. Since every sub-segmentation needs to follow the harmonic-rhythm constraint (Harmonic Rule 2: Every chord label cannot sustain through a metric position stronger than where it starts—i.e., harmonic rhythm cannot be syncopated.), sub-segments "12.." and "12.3" will be eliminated since their harmonic rhythms are syncopated, leaving six legitimate sub-segmentations illustrated in Figure 5-3.
2. For each possible sub-segmentation, identify all notes that exhibit transitional motions—we call these "potential non-chord tones."
3. Compute every combination of potential non-chord tones, allowing that some interpretations are mutually exclusive (Melodic Rule 1: The antecedent and consequent note of each non-chord tone must be chord tones, except for the special case of double passing.).
4. For every legal combination of potential non-chord tones, remove these non-chord tones, group the remaining chord tones, and output the chord label if the quality is one of the triads or 7^{th} chords.
5. Discard interpretations whose chord qualities are not considered.
6. Discard incomplete harmonies when there are alternative, complete harmonies.
7. If the same chord is identified in two successive sub-segments (e.g., "1234" with chord labels "CGGC". The "G" chord is both identified in #2 and #3 segments.), discard this interpretation since an equivalent sub-segmentation that merges these two segments with the same chord can be found (e.g., "12.3" with "CGC").

8. If a slice is identified as a dyad/unison, and the preceding or succeeding slice is a superset of that dyad/unison, the slice is then associated with the label of the superset. For example, if the slice has a C-G dyad and the preceding or succeeding slice has a C major triad, the slice will be labelled as C major triad as well.

Figure 5-3 illustrates the application of this algorithm to the sixth window in the chorale shown in Figure 5-2. The four slices in this window can be divided into six sub-segments (Step 1), shown below the staff. Eleven of the twelve notes in the window are potential non-chord tones (labelled and enumerated in Figure 5-3). The algorithm tests various permutations of these potential non-chord tones (algorithm Steps 3–5) as follows: First, assume potential non-chord #1 is a non-chord tone and all other notes are chord tones. Under this assumption, segmentation 1... forms the illegal sonority {A,B,C,D,E,F♯}; segmentation 1.2. forms the illegal sonorities {B,C,D,E} and {A,B,C,E,F♯}; segmentation 1..2 forms the illegal sonority {B,C,D,E} and the legal sonority {A,C,F♯}; etc. Repeat this procedure for every other potential non-chord tone, every pair of non-chord tones, every triplet of non-chord tones, etc., while skipping mutually exclusive combinations—i.e., if #2 is an appoggiatura #4 must be a chord tone. Testing all non-chord tone permutations across all six segmentations reveals eleven non-redundant (Steps 6–8) interpretations with legal chords in all segments (Step 5). Of these eleven, we can "filter out" interpretations involving 7th chords (all the sevenths must be considered as non-chord tones, regarding Section 5.1.1.1), leaving the four triadic analyses shown in Figure 5-3. We can reduce the number of analyses by further specifying other preferences: The number will reduce to two if we prefer the fewest number of chords, leaving only the "C" chord and the "Am" chord interpretations for the entire window, and we will only have the "C" chord interpretation if we prefer the analysis with the most non-chord tones (regarding Section 5.1.1.2).

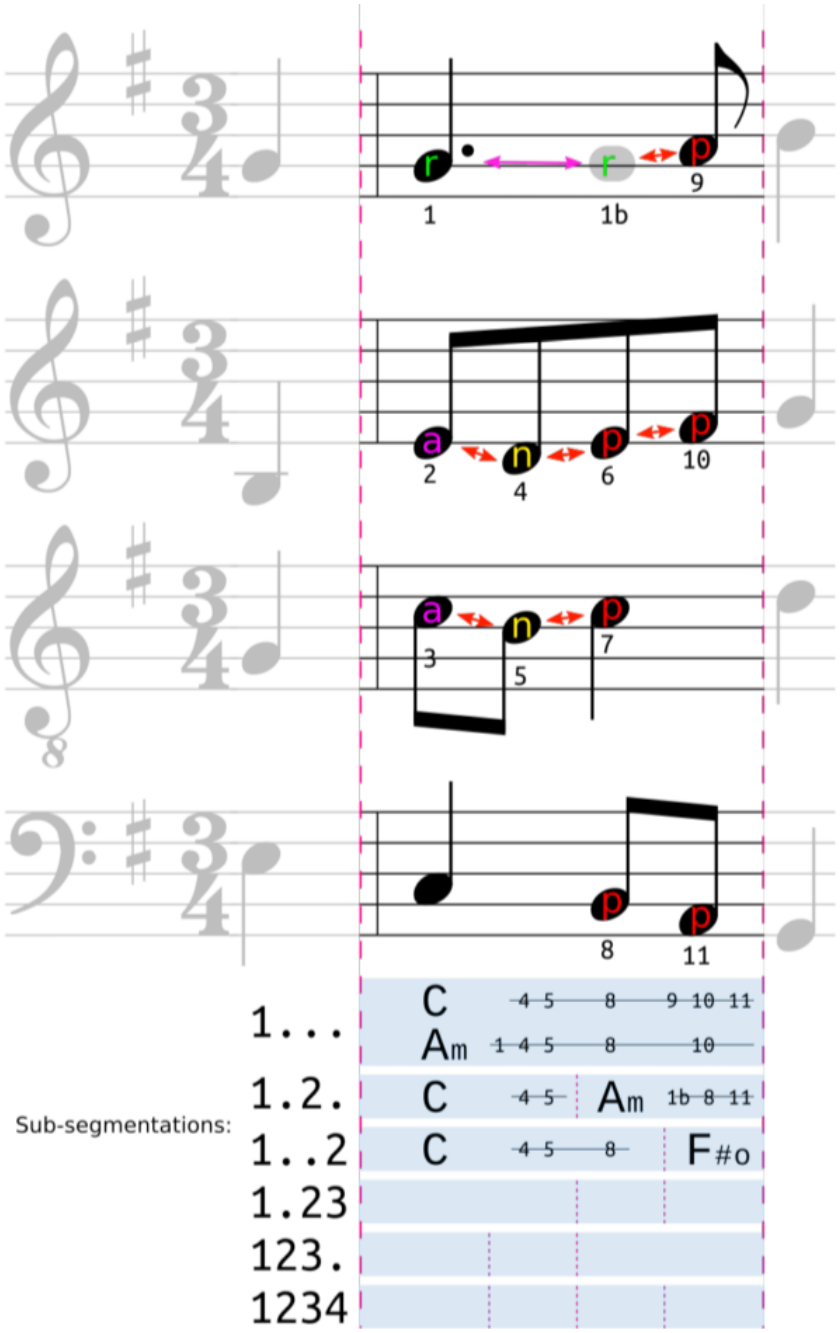


Figure 5-3: Illustration of the permutational analysis of a single contextual window (window 6 from Figure 5-2). Each note in the window is annotated as a potential non-chord tone, marked "p" for passing tone, "n" for neighbour tone, "r" for retardation, or "a" for appoggiatura—mutually exclusive potentials are annotated with arrows. The single unlabeled C must be a chord tone, as it does not exhibit any of the transitional motions. Below the staff, the six possible rhythmic segmentations of the window are shown. The four possible purely triadic interpretations of the window are shown; the notes interpreted as non-chord tones are identified (by number) beside each analysis.

5.1.3.5 Unusual cases

Chorale music is valued pedagogically for its simplicity and consistency. Nonetheless, a handful of chorales contain unusual features that complicate the batch analysis of the corpora. Notable examples in the Bach chorales include:

- An unusual call and response between the soprano and the rest of the voices in BWV 8;
- dissonant notes which resolve across phrase boundaries (i.e., through a fermata) in BWV 298, 407, and 320;
- suspensions which resolve indirectly in BWV 267 and 343.

A number of Praetorius chorales also contain subsections in which a subset of voices sings while the others rest, confounding our windowing heuristic. Solutions to these special cases, and a handful of others, were hard-coded into the workflow. Overall, it is important to note that all the rules in this algorithm, especially the hard-coded ones introduced above, are designed specifically for these 371 Bach chorales and 200 Praetorius chorales, therefore they will not generalize to other kinds of music. We (Condit-Schultz, Ju, and Fujinaga 2018) are aware of this limitation and will optimize the algorithm so that it can first be applied to other Baroque homorhythmic chorales, then perhaps to homophonic music in the next iteration.

5.1.4 Summary

The empirical and computational study of harmony is essential to furthering our understanding of musical structure and perception. However, this research must remain cognizant of the subtle complexities and controversies of harmonic theory if it is to be fruitful. In this section, we (Condit-Schultz, Ju, and Fujinaga 2018) presented a dataset of 571 Baroque homorhythmic chorales: The 200 newly encoded Praetorius chorales and the existing 371 Bach chorales, with chord label annotations generated by our rule-based algorithm which is not limited to one specific set of theoretical assumptions, allowing for just such subtleties to be explored systematically.

The dataset, the annotations, and the rule-based algorithm are publicly available at: https://github.com/DDMAL/Flexible_harmonic_chorale_annotations. To generate chord labels, users need to follow the installation guidelines (README) of the repo, deploy the algorithm

locally, and read the documentation to use the API (Application Programming Interface), which enables them to generate analyses of one particular preference using a set of filters. The documentation contains an exhaustive list of filters and usage examples, thanks to Nathaniel Condit-Schultz's dedicated work.

We hope that this dataset and this rule-based algorithm will facilitate research into tonality and harmonic progression, especially changes in harmonic practice between the early 1600s and the mid-1700s. However, our grander purpose is to facilitate a critical, data-driven approach for automatic chord labelling, which will be introduced in Section 5.2.

5.2 The interactive workflow

In this section, I will propose an interactive workflow for automatic chord labelling. First, I will use the rule-based algorithm proposed in Section 5.1 to generate preliminary, theoretically consistent chord labels by specifying a combination of filters as a specific analytical perspective (Section 5.2.2.3). The resulting annotations will contain single labels for chords, which will be used to pre-train three machine learning models. These four models (the rules-based model and the three models trained with machine learning) are grouped into an ensemble that generates chord labels by voting, and then a domain expert manually corrects only those chords that the ensemble does not agree on unanimously. Finally, I use these corrected annotations to re-train the machine learning models as our final automatic chord labellers, evaluated on a test set fully annotated manually by an expert. The work introduced in this section has been published at the 20th International Society for Music Information Retrieval Conference (Ju et al. 2019). Therefore, most of the text in the following sub-sections (within Section 5.2) is used verbatim without using quotation marks from this paper. The ground truth for the 39 Bach chorales introduced in Section 5.2.3.1 was prepared by the co-author Samuel Howes.

5.2.1 Introduction and basic methodology

In Section 5.1, a rule-based algorithm is proposed to generate variant analyses based on different analytical strategies. One general distinction among these analyses is that, some analyses contain fewer chord changes, while others contain more chord changes. We characterize them as "melodic" strategy and "harmonic" strategy, respectively, which is illustrated in Figure 5-4.

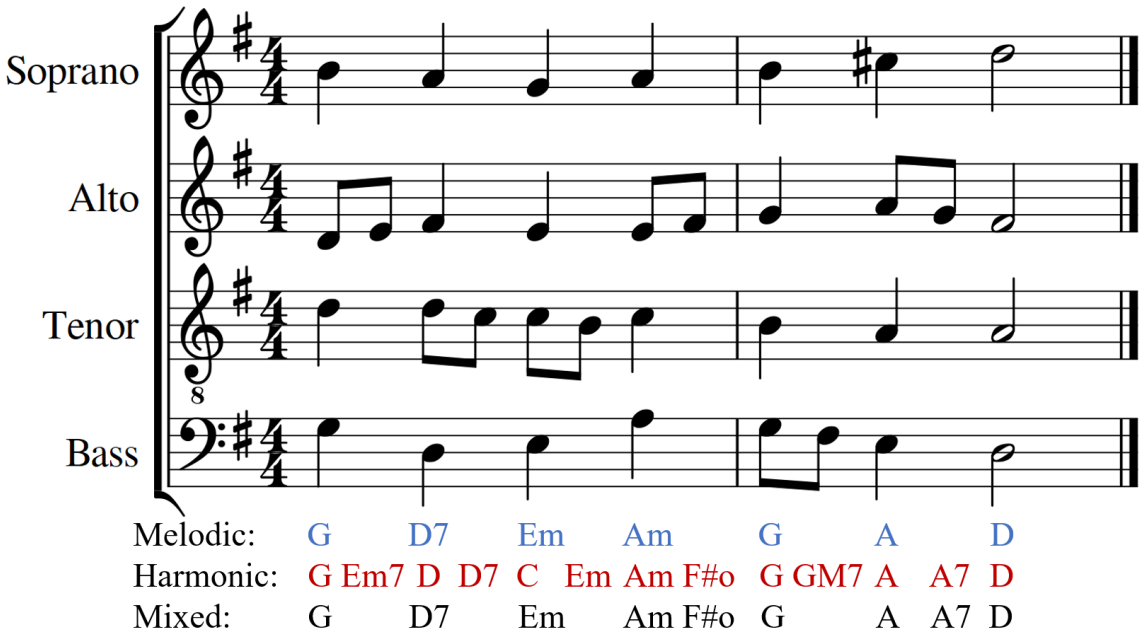


Figure 5-4: A sample chorale passage Nathaniel Condit-Schultz (co-author of the paper Ju et al. (2019)) composed and annotated with important differences between melody-oriented (blue) and harmony-oriented (red) analyses. The final analysis (black) mixes the two styles. Such inconsistencies are quite common, even between expert analyses.

This rule-based approach has been used in the past for automatic chord labelling (Temperley 1997; Temperley and Sleator 1999; Temperley 2001; Winograd 1968; Maxwell 1992; Pachet 1991; Scholz, Dantas, and Ramalho 2005; Hoffman and Birmingham 2000; Tojo, Oka, and Nishida 2006; Rohrmeier 2006; De Haas 2012; Granroth-Wilding 2013; Barthélemy and Bonardi 2001; Pardo and Birmingham 2002). Although this approach will generate chord labels that are internally consistent, they often fail to produce correct analyses for even moderately exceptional passages, as it is extremely complicated to define rules that are comprehensive enough to account for all possibilities.[36]

Other researchers have made use of manual annotations by experts, who can better respond to exceptions (Burgoyne, Wild, and Fujinaga 2011; De Clercq and Temperley 2011; Devaney et al. 2015; Hedges and Rohrmeier 2011; López 2017; Neuwirth et al. 2018). Such ground truth can be used to train machine learning (ML) models for automatic chord labelling (Koelsch et al. 2004;

[36] This limitation was also acknowledged in the beginning of Chapter 5, where our rule-based algorithm was first mentioned, and we referred to the generated analyses as *preliminary*.

Raphael and Stoddard 2004; Patel 2010; Tsui 2002; Kröger et al. 2008; Mearns 2013; Masada and Bunescu 2019; Chen and Su 2018; 2019; Micchi, Gotham, and Giraud 2020). Although the annotations created by human analysts are more nuanced, manual harmonic annotations require an enormous amount of time and expertise, and can be inconsistent (Ni et al. 2013; Koops et al. 2019), which may undermine an ML model's effectiveness, especially when limited amounts of training data are available.

Due to these difficulties, few large high-quality datasets and automatic chord labelling models exist, a situation that has significantly limited the computational study of Western harmony. In this section, we combine the strengths of existing approaches to address common problems of automatic chord labelling within a single interactive workflow, using a set of largely homorhythmic (see Glossary) Bach chorales. The proposed workflow is illustrated in Figure 5-6 and described below:

- To solve the problem of analytical inconsistency, we use the rule-based (RB) model introduced in Section 5.1 to generate preliminary, consistent chord labels according to a particular analytical style, resulting in only single labels for chords.
- These analyses are used to pre-train three ML models,[37] which together with the RB model form an algorithm ensemble, where each model within the ensemble labels all the chords. The most-preferred chord labels[38] are then output as *Analysis 1*.

[37] See the caption of Figure 5-6 for the details of these models.
[38] If there is a tie, prefer the label for which the rule-based algorithm voted. This is done for simplicity, where other preferences can also be experimented with in the future.

- To improve the quality of the analyses, a human expert examines only the onset slices for which the generated chord labels within the ensemble did not agree unanimously.[39] These slices will be highlighted in scores, where the human expert will analyze the music (without seeing any chord labels output by the ensemble) and provide manual chord label annotations that replace those of *Analysis 1*. We call this process "partial manual modification", with the example shown in Figure 5-5. Compared to annotating chorales from scratch, the amount of required work for the expert is reduced. The first three steps of this workflow are shown in Part 1 of Figure 5-6.
- Once the expert's corrections are obtained (Analysis 2), we will re-train the ML models. The most-preferred chord labels from the new ensemble are chosen as the final chord labels (Analysis 3), which is shown in Part 2 of Figure 5-6. This paradigm of manually modifying the generated data and re-training the ML models is known as "interactive machine learning" (Amershi et al. 2014; Fails and Olsen Jr 2003).

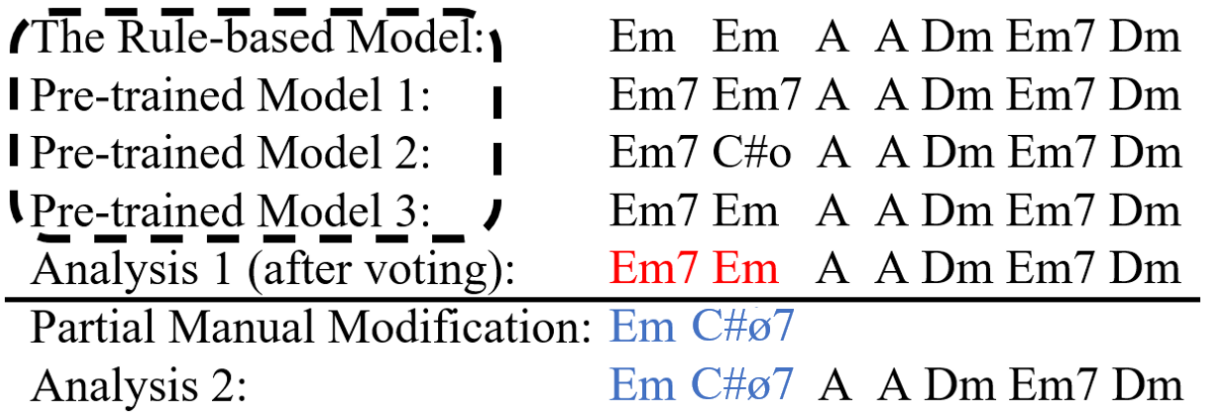


Figure 5-5: Illustration of partial manual modification, where the dashed rectangle indicates the algorithm ensemble, with their generated chord labels shown on the right. The chords in red are the ones with which the ensemble did not agree unanimously. The human expert analyzes the music of these areas and provides manual annotations (in blue) that replace the original chords (in red) and form Analysis 2, which is used for retraining as shown in Part 2 of Figure 5-6.

[39] We are aware that even if all chord labels within the ensemble agree unanimously, there is still chance of oversight that these chord labels can be wrong, since the ensemble is built on the preliminary chord labels generated by the RB model without external experts' validations.

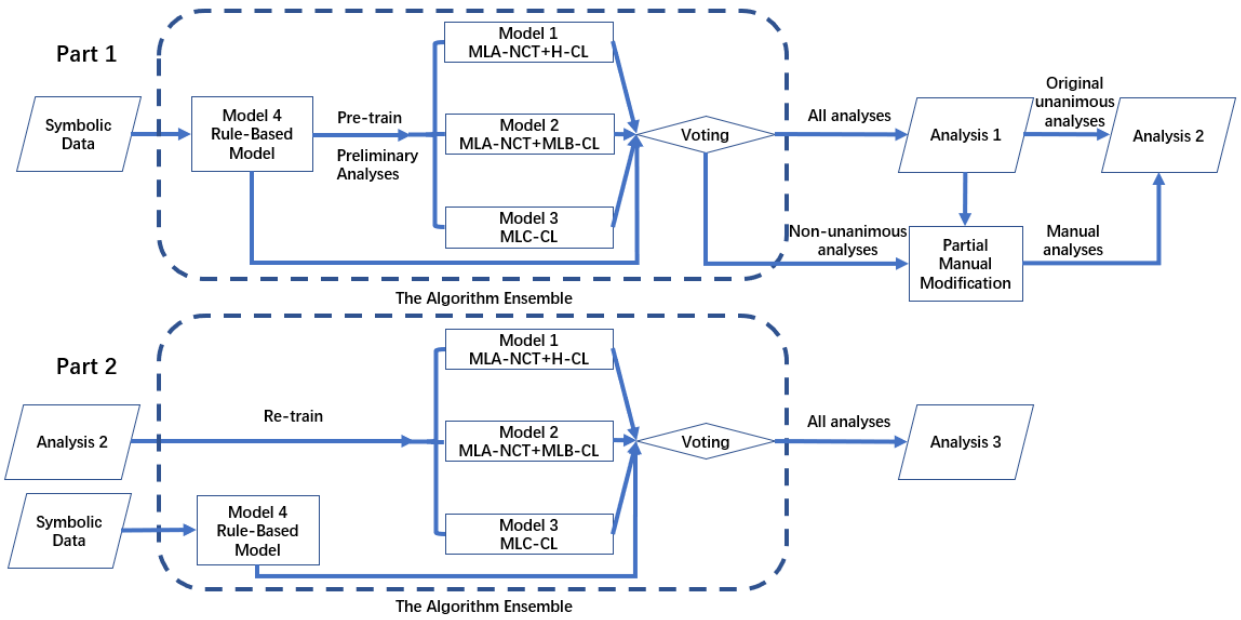


Figure 5-6: Interactive workflow for automatic chord labelling. There are four models within the algorithm ensemble, three of which are trainable. Models 1 and 2 both use a machine learning algorithm (MLA) to identify and remove non-chord tones (NCTs). After this, Model 1 (MLA-NCT+H-CL) uses a heuristic (H) algorithm and Model 2 (MLA-NCT+MLB-CL) uses an ML algorithm (MLB) to infer chord labels (CL) from the remaining chord tones. I term this process "NCT-first chord labelling", as shown on the right side of Figure 5-7. Model 3 (MLC-CL) uses a single ML algorithm (MLC) to infer chord labels (CL) directly from the pitch class collections, without removing NCTs. I term this process "direct chord labelling", as shown on the left side of Figure 5-7.

This workflow is not limited to Bach chorales. With an adapted RB model (introduced in Section 5.1, Model 4 in Figure 5-6), it can easily be applied to other genres of music in a fully automatic way (ending with Analysis 1) or interactively if an expert analyst is available (ending with Analysis 3).

5.2.2 Details of the methodology

This section introduces additional details of the interactive workflow shown in Figure 5-6 and described in Section 5.2.1.

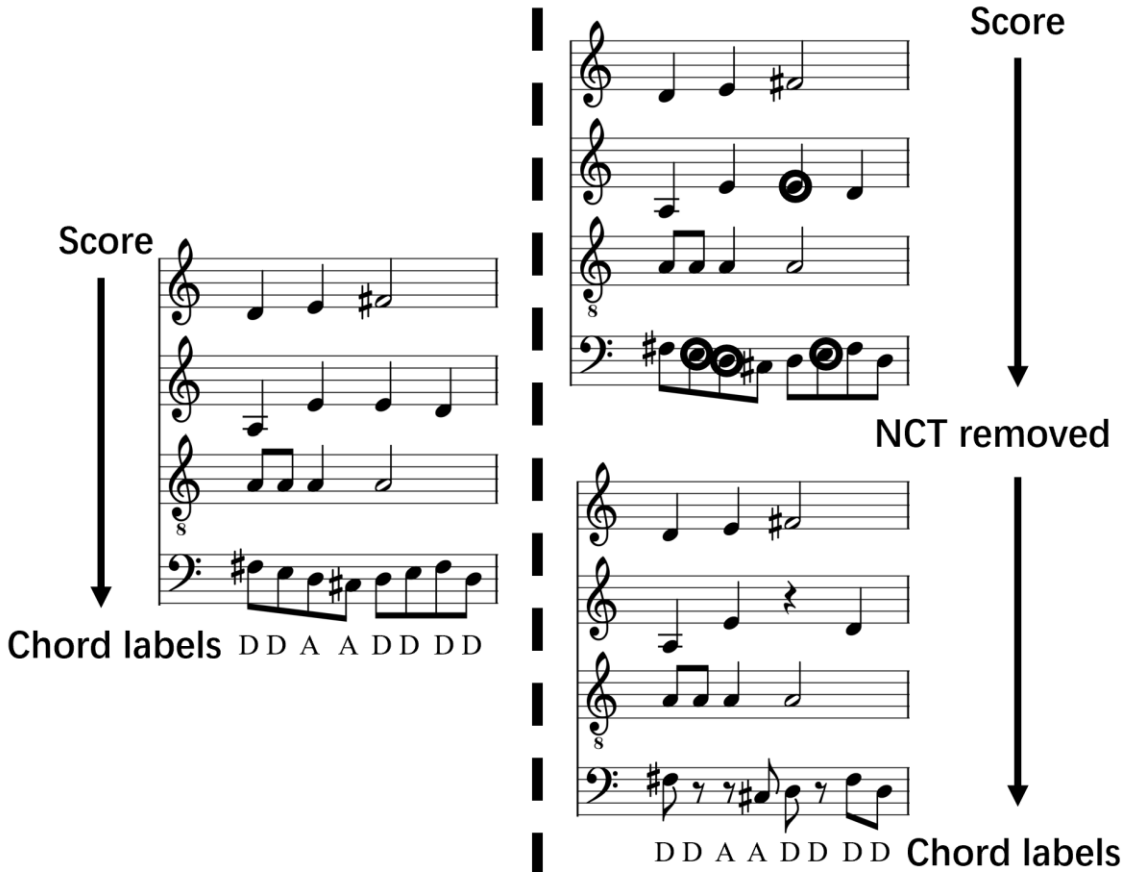


Figure 5-7: Comparison of "direct chord labelling" (left, used by Model 3 in Figure 5-6) and "NCT-first chord labelling" (right, used by Model 1 and Model 2 in Figure 5-6) approaches to automatic chord labelling. The former identifies chords directly from the score, while the latter first identifies and removes non-chord tones from the score, and then generates chord labels from the remaining chord tones.

5.2.2.1 Input data encoding and processing

The workflow currently accepts music encoded in Humdrum's **kern symbolic representation. Any other formats that can be faithfully converted to **kern can also be used. Each chord label consists of the letter-name of the root and the quality of the chord (e.g., C major). Triads can be major, minor, or diminished; 7th chords can be major, minor, dominant, half-diminished, or fully diminished. Chord letters (Section 3.2.3) are used to indicate chord labels, and chordal inversions are not specified.

Chord labels are appended to the original **kern file for each chorale and aligned with the music as "note onset slices", as illustrated in Figure 5-8. An onset slice is formed whenever a new

note onset occurs in any musical voice, and consists of a list of all pitch classes sounding at that moment. Additionally, all chorales and corresponding chord labels were transposed to the same key to make the tonal relationships between pitch classes consistent across the dataset.[40]

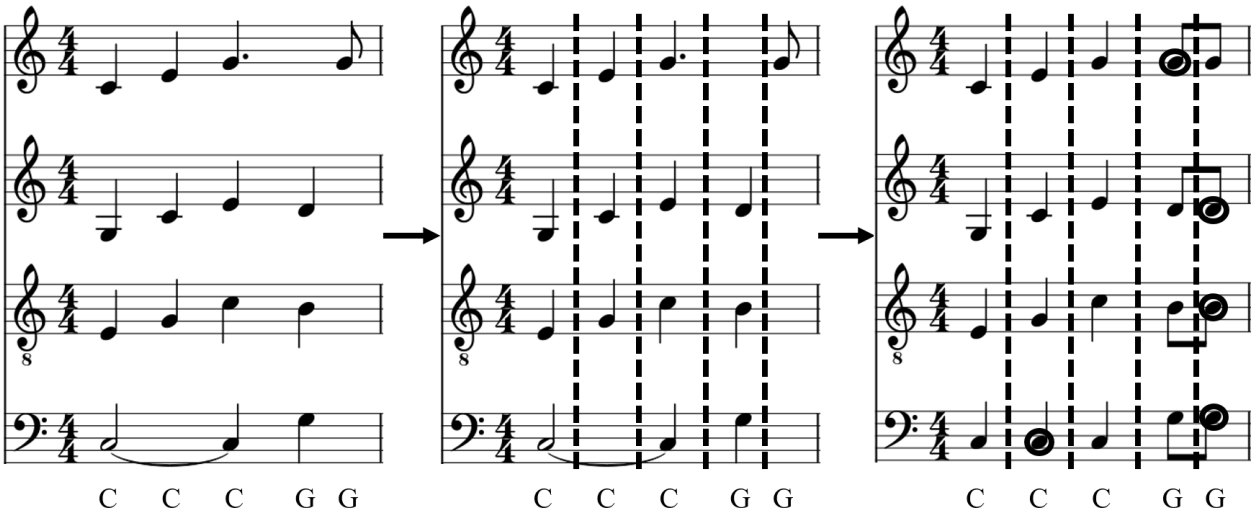


Figure 5-8: Illustration of note onset slices, aligned with chord labels. An onset slice is created whenever a new note onset occurs in any musical voice (middle). Any note sustained from a previous slice becomes an "artificial onset" in the new slice (right, circled).

5.2.2.2 Input features

Each onset slice is mapped to a feature vector for processing by Model 1, Model 2, and Model 3 of the workflow (see Figure 5-6). These features, and the codes used to refer to them in Section 5.2.3, are as follows:

1. **PC12**: A 12-D binary vector of enharmonic pitch classes present in the slice.
2. **M**: A 3-D indication of the metrical context of the slice, which specifies whether a slice occurs on the downbeat of a measure, on another whole beat (e.g., beat 2, 3, or 4 in 4/4), or on a fractional beat (e.g., beat 3.5).
3. **O**: A 12-D vector indicating which PC12 pitch classes are real onsets and which are artificial onsets (see Figure 5-8).

[40] The built-in key transposition function from music21 was used, with the Aarden-Essen key profile (http://web.mit.edu/music21/doc/moduleReference/moduleAnalysisDiscrete.html#aardenessen) . Chorales were transposed to C major or A minor, depending on their modes.

4. **Wn**: A variable size vector containing the (non-Wn) features from the n previous and n following slices (e.g., W1 indicates that features for the directly preceding and directly following slices are included in the features of the current slice). These surrounding slices are called "contextual windows."

Each of these feature categories used a one-hot or multi-hot encoding schema, where M uses the former that only has one bit activated, while PC12 and O use the latter that can have multiple activated bits. The workflow allows for experimentation with different feature subsets and values of n. For example, a "PC12M" configuration indicates a 15-D vector, with O and Wn features omitted. This notation is adopted in Section 5.2.3.

5.2.2.3 Rule-based algorithm

I use the RB algorithm proposed in Section 5.1 to generate preliminary chord labels (Model 4 in Figure 5-6). A "harmonic" rather than "melodic" style of analysis is used (see Figure 5-4), which prefers fewer non-chord tones (NCTs), and is better suited to the typical chorale texture. Below is the overview of this analytical perspective, which contains five ordered filters.[41] These five ordered filters are used to form a specific analytical strategy, so that there is only a single label for each chord.

- Prefer chords whose durations are no less than half a beat. Without this filter, the model will occasionally arrange new chords on the 16th beat, which is unnecessary in homorhythmic music.
- Prefer seventh chords only when the seventh note is resolved properly (down by a step).
- Prefer chords with the fewest NCTs. As shown in Figure 5-1 and discussed in Section 5.1.1.2, many notes in the musical surface can either be identified as CTs or NCTs. By preferring the fewest NCTs, most chords will only have one definitive label.

[41] To replicate these chord labels, you can deploy the rule-based algorithm locally and follow the installation guidelines at: https://github.com/DDMAL/Flexible_harmonic_chorale_annotations/releases/tag/v0.1.1, the README file within the source code.
To understand the syntax and usage of these filters, please refer to the documentation at: https://github.com/DDMAL/Flexible_harmonic_chorale_annotations/blob/master/FlexibleChoraleHarmonicAnalysis_0.8.0.pdf.

- In case there are multiple analyses with the same number of NCTs, prefer the one with the fewest chords. Since it prevents unnecessary chord changes, especially when a triad has a delayed seventh note. Instead of considering a triad changing into a seventh chord, the whole section will be labelled with a seventh chord.
- If there are multiple analyses with the same number of NCTs and chords, prefer chords with the minimum delay of completion (e.g., all the chord tones of the chord label should appear as soon as possible).

For Model 1, we also used a heuristic algorithm (H-CL from Figure 5-6) to infer chord labels from remaining chord tones. The details of this algorithm are shown below:

- All the slices will be iterated first, and for those chord tones that comprise a chord quality considered in Section 5.2.2.1, the slice will be annotated with this chord. Otherwise, the slice will be labelled as "Undetermined".
- All the "Undetermined" slices will be iterated for the second time. For each of these slices, we will choose from the closest preceding or following chord label whose chord tones overlap the most with those in the "Undetermined" slice. For example, if the current slice is "undetermined" with the pitch class collection CFG, the preceding chord label being C major, and the following chord label being A minor, C major will be chosen for this "undetermined" slice since C major overlaps more with CFG (C and G) than A minor. If there is a tie, choose the chord label whose root is the bass of the "Undetermined" slice.

- At this point, every slice will have a chord label, and we finally iterate all the slices with only two pitch classes, which form an interval that is a minor third, major third, perfect fifth or a tritone apart. For these slices, we first examine whether the associated chord label fully covers both notes of the interval. If not, we will label this slice with a new triad using the following rules:
    - If the interval is a minor third, we add a perfect fifth from the bass, forming a minor triad.
    - If the interval is a major third, we add a perfect fifth from the bass, forming a major triad.
    - If the interval is a perfect fifth, we add a major third from the bass, forming a major triad.
    - If the interval is tritone apart, we add a minor third from the bass, forming a diminished triad.

5.2.2.4 Machine learning algorithms

As shown in Figure 5-6, the workflow includes three ML algorithms (MLA, MLB, and MLC) to pre-train. MLA identifies NCTs from the musical surface, and this work has been published in the 4th International Workshop on Digital Libraries for Musicology (Ju et al. 2017). The output of MLA is a 12-dimensional vector specifying which pitch classes are both present and identified as NCTs; MLB and MLC label chords based on the musical content; they output similar vectors identifying the predicted single chord label among all candidates.

I experimented with both Support Vector Machines (SVMs, Cortes and Vapnik 1995) and Deep Neural Networks (DNNs, fully-connected feedforward neural networks, Ivakhnenko and Lapa 1965) in the implementations MLA, MLB, and MLC classifiers. For the DNN we used three hidden layers, each with 300 hidden units. Adaptive Moment Estimation was used as an optimizer, with loss functions of binary cross-entropy for MLA and categorical cross-entropy for MLB and MLC. The SVMs used linear kernel functions.

5.2.3 Experiments

5.2.3.1 Data

The experiments below were performed on a modified[42] dataset of Bach chorales originally produced by Craig Sapp.[43] This modified dataset consists of 369 homorhythmic chorales, where Chorale 150 is a five-voice chorale and the rest are all four-voice chorales.

To evaluate the performance of our workflow, 39 chorales were randomly chosen before the experiments began and partitioned into a set reserved for final testing in Experiment 2. These reserved chorales had their chords hand-labelled in their entirety by Samuel Howes, a music theory Ph.D. student from McGill University.

The remaining 330 non-reserved chorales were used for training, validation (early-stopping[44]) and internal testing.[45] The initial (potentially flawed) "ground truth" for these remaining 330 chorales consisted of the labels predicted by the RB model (Model 4, from Condit-Schultz, Ju, and Fujinaga (2018)). This imperfect "ground truth" was used in Experiment 1 (see Section 5.2.3.2) to get a preliminary sense of how well the workflow's component classifiers performed. The final evaluation was performed in Experiment 2 (see Section 5.2.3.3) with the proper, hand-annotated 39-chorale reserved test set.

In the following two experiments introduced in Section 5.2.3.2 and Section 5.2.3.3, we will use the *independent two-sample t-test*,[46] which compares one variable (i.e., whether the use of a certain feature introduced in Section 5.2.2.2 results in a significant change in performances)

---

[42] Available at: https://github.com/DDMAL/Flexible_harmonic_chorale_annotations/tree/master/kernData. Some corrections were made to the music and Chorale 150 was added to the dataset. Chorales 130 and 316 were excluded, since the original **kern files and the music21-parsed results are different.

[43] https://github.com/craigsapp/bach-370-chorales.

[44] Early stopping is a form of regularization used in machine learning to avoid overfitting. The training process will be terminated if the loss on the validation set does not decrease for a certain number of epochs.

[45] "internal testing" means this test set does not contain human annotated labels, and is used mostly for model selection, as discussed in Section 5.2.3.2. Since ten-fold cross validation is used, this internal testing set therefore covers all the 330 non-reserved chorales across all folds. Here, the hyperparameters of DNN were tuned using this internal testing set. Different kinds of kernel functions for SVM were also experimented with using this internal testing set, and the linear kernel function was finally chosen (see Section 5.2.2.4). These hyperparameters were chosen since the resulting DNN and SVM respectively achieved the highest mean value of classification accuracy across all folds in the internal testing set.

[46] The tool available at: https://www.graphpad.com/quickcalcs/ttest1.cfm.

between two groups. This test makes three assumptions that are not necessarily met in the analysis performed here: (1) The two groups being compared are independent, (2) the samples in each group are normally distributed, and (3) the variances from two groups are considered homogeneous. The first assumption (independence) is not met, given that cross-validation is involved, and the remaining two assumptions have not been tested. We nonetheless used the *independent two-sample t-test* here as a preliminary approximation of statistically valid significance testing.

5.2.3.2 Experiment 1

Experiment 1 tested the effectiveness of several different workflow configurations by experimenting on varying input features and learning algorithms (see Section 5.2.2.4). The performance of Models 1, 2, and 3 from Figure 5-6 were tested.

5.2.3.2.1 Experimental setup

Ten-fold cross-validation is used to see the overall performance (average accuracy and standard deviation) of our model, which is being trained, validated, and tested on different portions of the dataset, specifically the 330 non-reserved chorales described in Section 5.2.3.1. For the DNN experiments, We divided the non-reserved portion of the dataset (330 chorales) into training (80%), validation (10%) and internal testing (10%) folds. For SVM, the data was divided into training (90%, the union of the DNN training and validation sets) and internal testing (10%, matching the DNN internal test sets) folds. When the W features were included (see Section 5.2.2.2), *n* was set to 1 for MLA and MLC, and 2 for MLB (represented as W1/2).

5.2.3.2.2 Results

The results of Experiment 1 are shown in Table 5-1. The highest classification value of 90.1% was achieved by Model 2 using the PC12MOW1/2 input features. Results show that the addition of a small contextual window (feature Wn) improved the performances of Model 2 and Model 3 with statistical significance.[47] This matches the general music-theoretical understanding

[47] $p<0.05$ in independent two-sample t-test comparing all Model 2 and 3 accuracies for PC12 and PC12M with those of PC12W1/2 and PC12MW1/2.

that, in cases of ambiguous harmony (e.g., an incomplete chord), a chord's immediate context is essential in labelling it properly.

It is important to note that these Experiment 1 findings are based on imperfect ground truth labels (see Section 5.2.3.1), and so must be interpreted as preliminary indications rather than as confirmed truth. Experiment 2 was performed to obtain more empirically meaningful results.

Table 5-1: Experiment 1 cross-validation classification accuracies, averaged across folds. Uncertainty values indicate standard deviation across folds. Values indicate the percentage of onset slices "correctly" classified by Model 1 (CA1), Model 2 (CA2), and Model 3 (CA3), based on the potentially imperfect Model 4 "ground truth". Columns indicate features (see Section 5.2.2.2) and rows indicate machine learning algorithms (see Section 5.2.2.4). The highest performance in each column is highlighted in bold.

Model	Metric	PC12	PC12M	PC12W1/2	PCMW1/2	PC12MOW1/2
SVM	CA1	**81.7±1.4%**	81.6±1.4%	82.7±1.0%	83.0±1.0%	83.5±0.9%
	CA2	73.0±1.5%	73.1±1.6%	85.4±1.3%	86.1±1.5%	87.4±1.5%
	CA3	74.9±1.6%	75.6±1.5%	85.4±1.3%	85.9±1.3%	87.7±1.5%
DNN	CA1	81.0±1.5%	**81.7±1.5%**	85.3±0.9%	85.6±0.9%	85.8±0.9%
	CA2	74.2±1.8%	75.1±1.6%	**88.5±1.3%**	**89.6±1.3%**	**90.1±1.5%**
	CA3	74.6±1.8%	75.3±1.4%	87.5±1.7%	88.3±1.7%	89.0±2.0%

5.2.3.3 Experiment 2

Experiment 2 compared the performance of the classifier ensemble after fully automated training (Analysis 1 in Figure 5-6) with that of the ensemble after human-assisted re-training (Analysis 3 in Figure 5-6). This set of experiments involved evaluation on a reserved expert-labelled test set (see Section 5.2.3.1).

5.2.3.3.1 Experimental setup

Classification models were pre-trained, had their outputs manually corrected, were re-trained on this corrected data, and then tested using the full workflow described in Section 5.2.1. Pre-training was done using the Model 4 output, just as in Experiment 1.

For the DNN training, we used 90% of the 330 non-reserved chorales as the training set and 10% as the validation set. A cross-validation-like training scheme was used: We conducted 10

experiments by training 10 models with rotated training and validation folds, while the testing fold (39 reserved chorales) remained the same. It is adopted to see the overall performance (accuracy and standard deviation) of our model trained and validated with different sets of data. All 330 non-reserved chorales were used to train each of the SVM classifiers. Only the PC12MOW1/2 input features (see Section 5.2.2.2) were used in Experiment 2, since they performed best in Experiment 1. For the W features, *n* was set to 1 for MLA and MLC, and 2 for MLB (represented as W1/2). Additionally, we also implement data augmentation (abbreviated as A, adopted by the PC12MOW1/2A configuration specified in Table 5-2 by transposing each chorale to 12 possible keys, resulting in an augmented training dataset 12 times as big as the original one). It is tested on DNN since such deeper models should benefit a larger amount of training data (Salamon and Bello 2017; Sun et al. 2017).

Once Analysis 1 (see Figure 5-6) was obtained, the human expert manually corrected only those chords that the ensemble did not agree on unanimously. The corrected labels (Analysis 2) were then used to re-train Models 1, 2, and 3. The 39 manually labelled reserved test chorales were then used to test the original pre-trained models, and then the re-trained models. Performance on this reserved test set is shown in Table 5-2.

5.2.3.3.2 Results

One can see in Table 5-2 that the original RB algorithm (Model 4 in Figure 5-6) attains a chord accuracy of 90.7% on the reserved test set, which serves as our baseline. The highest accuracy obtained by the pre-trained ensemble is 91.4%, using PC12MOW1/2, SVM classifiers, and voting. This (pre-trained) performance is achieved without any expert human intervention. It is of interest that CAVote here is higher than CA4, even though the classifiers in CAVote were trained on the RB output; this is perhaps because the RB model is overfitting the theoretical model underlying it, and that the pre-trained ensemble trained on it may in fact be smoothing out some of this overfitting to result in a slightly more general model. A comparison of Table 5-1 and Table 5-2 indicates that the Table 5-1 performance with artificial ground truth is quite similar to the performance of the Table 5-2 pre-trained classifiers on the reserved test set; this encouragingly suggests that the rules-based implementation is accurately modelling the chord labels.

Table 5-2: Experiment 2 classification accuracies on the reserved test set. DNN values are averaged across models trained using different training/validation sets, and uncertainty values indicate standard deviation across these folds. Values indicate how many onset slices were correctly classified by Model 1 (CA1), Model 2 (CA2), Model 3 (CA3), Model 4 (CA4), the ensemble as a whole (CAVote), and just those CAVote predictions that were unanimous (PUA). "PC12MOW1/2" indicates the input features (see Section 5.2.2.2. "Pre-trained" indicates performance before manual correction (i.e., Analysis 1 in Figure 5-6), and "Re-trained" indicates performance after re-training on the corrected data (i.e., Analysis 3 in Figure 5-6). We also explore data augmentation (PC12MOW1/2A) for the re-trained DNN model. The best performance in each column is highlighted in bold.

Model	Metric	PC12MOW1/2 Pre-trained	PC12MOW1/2 Re-trained	PC12MOW1/2A Re-trained
SVM	CA1	85.9%	87.0%	
	CA2	88.6%	89.8%	
	CA3	87.7%	89.3%	
	CAVote	**91.4%**	92.7%	
	PUA	79.1%	79.0%	
DNN	CA1	85.4±0.2%	88.2±0.2%	88.6±0.2%
	CA2	88.9±0.3%	91.3±0.4%	93.0±0.3%
	CA3	87.9±0.7%	90.5±0.3%	91.6±0.3%
	CAVote	90.9±0.2%	**93.5±0.2%**	**94.0±0.2%**
	PUA	80.4±1.2%	79.7±0.4%	80.7±0.3%
RB	CA4	90.7%		

Table 5-2 also shows that performance improved after retraining in most cases.[48] The best-performing[49] configuration attains an accuracy of 93.5% (using voting) for DNN trained on PC12MOW1/2 features without data augmentation. It is also worth noting that the re-trained DNN model achieved 94.0% with data augmentation.

The partial manual modification workflow is also found to be relatively efficient, as the expert analyst was only required to provide manual analyses for about 20%[50] of all slices.

[48] $p<0.05$ in independent two-sample t-test comparing results before and after retraining for CA1, CA2, CA3, and CAVote, but not PUA.
[49] $p<0.05$ in independent two-sample t-test comparing results of CAVote to CA1, CA2, CA3, and CA4.
[50] This value is inferred from Table 5-2: 100% - PUA.

Compared to examining and annotating every slice, the amount of required work is reduced substantially.

5.2.4 Discussion

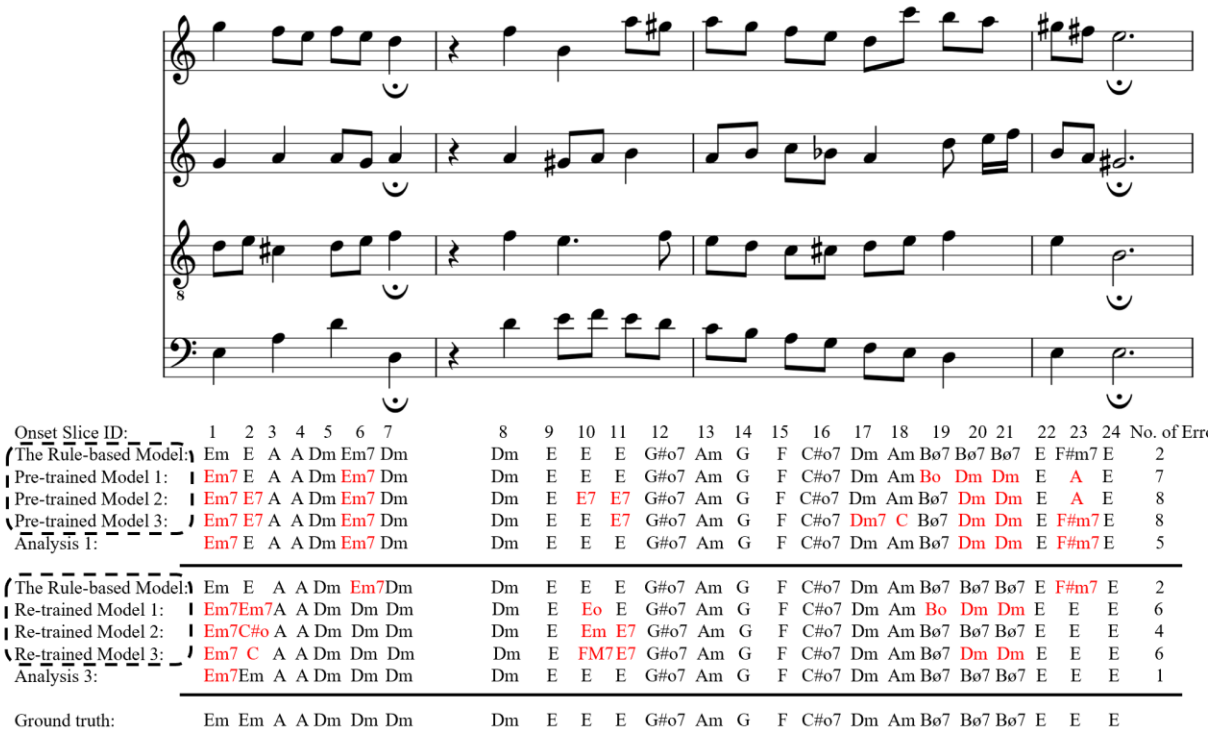


Figure 5-9: An illustration of how classifications evolve as processes proceed as outlined in Figure 5-6, based on measures 9 through 12 of BWV 315 "Gib dich zufrieden und sei stille." Chord labels were generated by a DNN-based algorithm ensemble using PC12MOW1/2 features (see Section 5.2.2.2). The algorithm ensemble comprises the four models within the dashed rectangle, which vote to generate Analyses 1 and 3. The labels above the first horizontal line were generated in a fully automatic way, without any human intervention. The labels between the two horizontal lines (other than the rule-based model) were generated automatically after re-training on partially manually corrected data. The chord labels highlighted in red are errors compared to the ground truth provided by an expert analyst.

According to the results, our interactive workflow performed well on the Bach dataset using a "harmonic" style of analysis. It was found that quite good performance could be achieved with our rule-based model (90.7% on the reserved test data), that performance could be improved slightly using the RB model to self-train a classifier ensemble (91.4% on the test data), and that still greater improvements resulted from partial manual modification with re-training (93.5% on the test data) and data augmentation (94.0% on the test data). Although these improvements may

seem small in absolute terms, they are statistically significant, and they represent meaningful relative improvements (decreases) in the error rate (drops of 7.5% comparing pre-trained CAVote to RB, 30.1% comparing re-trained CAVote (without data augmentation) to RB, 24.4% comparing re-trained CAVote (without data augmentation) to pre-trained CAVote, and 7.7% comparing re-trained CAVote without data augmentation and with data augmentation). Of particular importance, the RB model and the pre-trained ML model require no human intervention, and the retrained ML model requires much less expert labor than full manual annotation.

Figure 5-9 provides an illustration of how this approach can be effective, using an excerpt from one of the reserved test set chorales. Although some algorithms within the ensemble make errors, the re-trained ensemble ultimately generates answers closer to the expert-annotated ground truth in Analysis 3. Upon examining the errors, we find that some of them are reasonable alternative versions of the ground truth: Chords with the same roots, but with or without an added seventh (slices 1, 11, 17, and 19); or chords that are subsets of the ground truth chords (slices 20 and 21). As a result, some of the "errors" that the ensemble makes in this particular excerpt are in fact theoretically acceptable answers. This is encouraging, as it suggests that at least some of the "mistakes" made by the classifiers may not be mistakes at all. We still count them as mistakes, compared to the consistent analytical perspective adopted in this section.

5.2.5 Summary

In this section, we presented a versatile interactive workflow that generated chord labels for homorhythmic music. It can be used in a fully automatic way or, with a relatively small amount of intervention from an expert human analyst who corrects a small, automatically selected fraction of the generated analyses. A re-trained classifier ensemble with data augmentation can then be produced with even better performances. The source code, data, and results from this project can be found at: https://github.com/juyaolongpaul/harmonic_analysis/releases/tag/v1.1.

Results showed that this workflow is quite compelling: It combined the consistency of rule-based models with the nuance of manual analysis to generate relatively inexpensive ground truth for training effective machine learning models. The resulting classifier ensemble was able to automatically generate highly consistent chord labels, which can serve as invaluable resources for musicians, composers, and music researchers alike.

5.3 Conclusion

In this chapter, I proposed a method of generating single labels for chords, which were then used to build automatic chord labelling models using single-label learning. As an implementation, I first proposed a rule-based algorithm in Section 5.1 that generated variant chord labelling analyses based on different analytical perspectives. In Section 5.2, I first precisely defined an analytical strategy, so that the rule-based algorithm generated only single labels for chords. These preliminary annotations were later modified by one music theory expert with more nuance. The resulting modified annotations combined the consistency of the rule-based model with the nuance of manual corrections and were used to train and evaluate automatic chord labellers using single-label learning.

In Chapter 6, I will focus on obtaining parallel tracks of chord labels as multiple ground truths based on figured bass annotations and the musical surface, which will be used to explore multi-label learning and label distribution learning in Chapter 7, two supervised machine learning paradigms that enable automatic chord labellers to generate multiple parallel answers for each chord, either in the form of binary labels or a distribution of probabilities, respectively.

Chapter 6 Obtaining multiple ground truths of chord labels: A figured bass approach

As discussed in Section 4.3, my ultimate goal is to build automatic chord labellers that are able to generate multiple chord labels. Multiple parallel analyses are essential in training such models, and obtaining these annotations is not a trivial task. Although chord labels prepared by experts are high-quality, these manual annotations are extremely expensive and difficult to scale. In this chapter, I propose an innovative approach of obtaining multiple parallel analyses of chord labels based on figured bass. I will first introduce figured bass as well as its theoretical and historical significance (Section 6.1), and I will build a digital dataset called Bach Chorales Figured Bass dataset (Section 6.2), and it can be used to explore automatic figured bass annotation (Section 6.3), which will generate figures for those Bach chorales with no such annotations. Then, I will propose a series of four different rule-based algorithms that generate chord labels automatically, based on both the figured bass annotations and the musical surface (Section 6.4) and present the resulting Bach Chorales Multiple Chord Labels (BCMCL) dataset (Section 6.5). BCMCL will be used to explore multi-label learning and label distribution learning for automatic chord labelling, which will be introduced in Chapter 7. The research involved in Sections 6.1–6.3 has been published at the International Society of Music Information Retrieval Conference (Ju et al. 2020) and the Music Encoding Conference (Ju, Margot, McKay, and Fujinaga 2020b). The creation of the Bach Chorales Figured Bass (BCFB) dataset introduced in Section 6.2 was done by me, co-authors Sylvain Margot and Luke Dahn collectively. The research involved in Sections 6.4–6.5 has been published at the 7th International Workshop on Digital Libraries for Musicology (Ju, Margot, McKay, and Fujinaga 2020a). Therefore, most of the text in these sections is used verbatim without using quotation marks from these papers. The co-author Sylvain Margot offered essential insights and nuance on designing these rule-based algorithms and the discussions of figured bass. He also helped proofread the manuscripts of both papers.

6.1 Introduction

Figured bass is a type of music notation that uses numerals and other symbols to indicate intervals to be played above a bass note (Williams and Ledbetter 2001), and which can provide

insight on underlying harmonies. It was commonly used in Baroque music, and served as a guide for performance, especially for the instruments improvising the *basso continuo* accompaniment (e.g., harpsichord, organ, lute, etc.). Figure 6-1 shows an example of figured bass, as well as how a harpsichordist might realize the figured bass as an improvised accompaniment. Such realizations are not typically explicitly included in scores, as the musical tradition of the time left them to be improvised based on the skills and taste of the continuo player.

In this chapter, we will use the terms "figure", "figured bass", and "figured bass annotation". Their definitions are as follows:

- "Figured bass" is the combination of a bassline, numerals, and other symbols associated with it (usually written above or below the bassline).
- A "figure" refers to a numeral (e.g., 2 at m. 2.3 of Figure 6-1), possibly with accidentals (e.g., ♯4 at m. 2.3 of Figure 6-1) or slashes (e.g., 5 at m. 2.3 of Figure 6-1).
- A "figured bass annotation" (FBA) refers to all the figures stacked vertically with respect to a single note in the bassline. For example, the combination of the two figures $\frac{\sharp 4}{2}$ at m. 2.3 of Figure 6-1 is considered a single FBA.

To find out which notes figured bass specifies, take the first "6" (m. 1.2) of Figure 6-1 as an example: The bass is B♭ and the diatonic notes of the current key signature are: [G, A, B♭, C, D, E♭, F], where G forms a 6th interval above the bass and thus is the note figured bass specifies. Note that figured bass does not indicate the specific quality of an interval: It is a major 6th interval in this case, and it can be a minor 6th interval if, for example, the bass is G and the "6" above the bass is E♭ in this key signature.

Also note in Figure 6-1 that the harpsichord also includes the note D (the upper staff of the Harpsichord part at m. 1.2) as part of the improvisation. This is because figured bass does not always specify all the notes for improvisation, and often omits some obvious figures (see Section 6.3.2.2 and Figure 6-3 for the specific rules of omission). The basic three aspects of figured bass have been introduced in Section 3.2.1, here I will re-iterate them as follows:

1. The slashed FBAs indicate altered intervals. FBAs consisting of numbers with backslashes through them indicate raised intervals (e.g., m. 3.3 and m. 3.4), and forward slashes indicate lowered intervals (e.g., 𝄫), where the corresponding notes are respectively raised or lowered by one semitone.
2. FBAs followed by continuation lines indicate that the harmony of the preceding figure is prolonged (e.g., m. 1.4, m. 4.2, and m. 4.4).
3. Multiple FBAs over a stationary bass (e.g., 4–3 in m. 5) may[51] indicate a suspension being resolved.

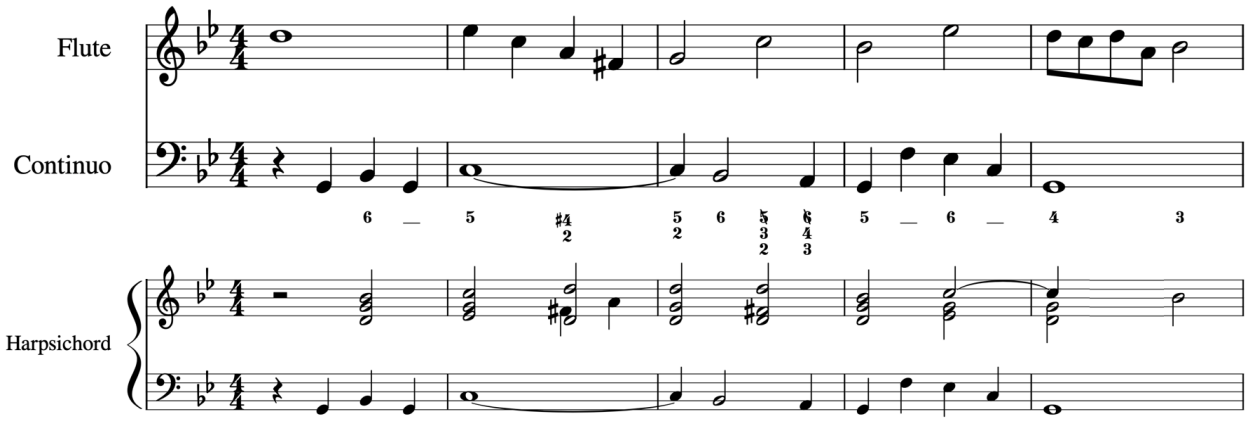


Figure 6-1: A sample musical passage the co-author Sylvain Margot composed, where figured bass annotations (FBAs) are shown below the continuo line, and where we added the harpsichord line as an example of what a continuo player might improvise based on the bass notes and accompanying annotations in the continuo line (typically, a score would only explicitly contain the continuo and flute parts, so we attached the harpsichord part as a separate system). Figures indicate intervals above the continuo line that could be played in the improvisation. For example, the "6" in the first measure corresponds to the pitch class "G", which is a 6th above the bass "B♭". An actual improvisation would likely also typically contain the pitch class "D" (a 3rd above the bass "B♭") in this slice, but this is not explicitly indicated in the figures. This is an example of how FBAs do not always specify all the notes to be played by the continuo player, and usually omit some obvious figures (see Section 6.3.2.2 for details).

Figured bass also serves pedagogical and theoretical purposes: Not only does it provide contrapuntal information on how to conduct the resolution of dissonances (e.g., 4–3 in m. 5 of Figure 6-1), but it also offers insights into the chords and harmonic rhythm intended by composers.

[51] To confirm a suspension, one must examine the musical surface to see whether the suspension is prepared properly.

Figured bass can therefore provide a preliminary description of harmonic structure and serves as a promising basis for approaching chord labelling.

As a useful analytical tool for studying Baroque compositional and performance practices, figured bass has been an important topic in music pedagogy (Bach [1753] 1949), music theory, and musicology (Remeš 2019). The computational study of figured bass, however, has drawn little attention over the years. We have only found two papers on automatic figured bass annotation, both using a rule-based approach: Barthélemy and Bonardi (2001) treated figured bass as a harmonic reduction and devised rules to identify and remove ornamental notes, permitting them to cluster the remaining chord tones as figures; and Wead and Knopke (2007), in contrast, manually designed a decision tree to determine the figured bass for a given bass line. Unfortunately, with no open-source code and a lack of quantitative results, it is impossible to objectively evaluate or compare the performances of these models. Furthermore, we are not aware of any previous applications of machine learning to figured bass, nor of any existing digital dataset with figured bass annotations (FBAs). These limitations have likely limited the computational study of figured bass to date.

6.2 Building the Bach Chorales Figured Bass dataset

To the best of our knowledge, there is no prior publicly available digital figured bass dataset. We therefore present the Bach Chorales Figured Bass dataset (BCFB), a corpus we constructed containing both the musical surface and FBAs in MusicXML, **kern, and MEI (Music Encoding Initiative) formats. It consists of all 139 J. S. Bach four-voice chorales that include his own figured bass, based on the Neue Bach Ausgabe (NBA) critical edition (J. S. Bach, Dürr, and Neumann 1954–2007). NBA was chosen as the source of BCFB because it is the most up-to-date scholarly critical edition. In Section 6.2.1, we will first find all the chorales with FBAs in NBA, then digitize these FBAs (Section 6.2.2) in the MusicXML format and provide these encodings in other symbolic formats (i.e., **kern and MEI, see Section 6.2.3).

6.2.1 Finding Chorales with figured bass annotations

To find all the chorales attributed to Bach, the co-author Luke Dahn constructed a reference table (http://www.bach-chorales.com/BachChoraleTable.htm) with all 420 chorales indexed by

BWV catalogue numbers, and cross-referenced them with the NBA. We checked whether original FBAs are accessible for each of these chorales, and found 139 settings meeting this criterion. We then made an expanded reference table, consisting of the: BWV number,[52] Breitkopf number (when relevant),[53] title of the work of origin (e.g., cantata, passion, etc.), date of the first performance, text setting, location of the score in the NBA edition, and other musicological metadata for each chorale. This table is designed to facilitate musicological research, which, along with the BCFB dataset, is available at: https://github.com/juyaolongpaul/Bach_chorale_FB.

6.2.2 Digitization

We began the creation of BCFB by assembling existing symbolic encodings of the relevant Bach chorales from the KernScores repository (kern.ccarh.org), which is maintained by Stanford's Center for Computer Assisted Research in the Humanities, and includes 370 four-part chorales (compiled by Craig Sapp and were also available at https://github.com/craigsapp/bach-370-chorales) encoded in the Humdrum **kern representation (www.humdrum.org). These chorales do not have FBAs, and we found 109 chorales matching the 139 NBA chorales with FBAs. We chose MusicXML as our master file format since it is widely supported by music notation software. We automatically translated these 109 **kern files into MusicXML using *music21* (v. 5.1.0), and then used the MuseScore (v.3.3.2) editor to make changes to match the musical content[54] of the NBA edition and add Bach's FBAs. We manually encoded the remaining 30 figured chorales found in the NBA edition as MusicXML files. As a result, we have a total of 139 chorales that contain both the music and the FBAs from NBA, digitized in the MusicXML format.

6.2.3 Converting to other symbolic file formats

In this section, we will introduce the methodology of converting the BCFB dataset (from MusicXML) to two other symbolic file formats: **kern and MEI, using the existing, most up-to-

[52] Bach-Werke-Verzeichnis (BWV) catalogue number, which indexes all the compositions attributed to J. S. Bach.
[53] The Breitkopf edition contains 371 four-voice J. S. Bach chorales, and indexes them differently from BWV.
[54] Including: adding a continuo line and/or instrumental voices; transposing; changing the meter, pitch, and duration of certain notes; etc. We did not encode the textual content specified in the NBA.

date software based on the insights of Nápoles López, Vigliensoni, and Fujinaga (2019).[55] This diversity of symbolic formats offers researchers the opportunity to use the format most convenient to their preferred software, because if only one format were offered, which might not be supported by a given piece of preferred research software, then it would need to be converted to the format supported by the software. This could lead to a potential loss of figured bass information or to other conversion errors (Nápoles López, Vigliensoni, and Fujinaga 2018). We also conducted a series of experiments on how much figured bass information is preserved when converted from one symbolic file format to another. For this purpose, we chose BWV 33.6[56] as the basis for a case study on how well figured bass notation can be encoded and translated, as it contains all three special aspects of FBAs introduced in Section 6.1. We also randomly sampled five other chorales and found no peculiarities in encoding and translation. However, we cannot guarantee that there are no such peculiarities for the rest of the chorales in BCFB. We used MuseScore (v.3.3.2) to encode the FBAs in MusicXML,[57] and a text editor for **kern[58] and MEI.[59] No problems were encountered during encoding, so these FBAs can serve as ground truth to evaluate the translated FBAs in each of the three formats.

Overall, there were some issues converting between the three formats. For the most part, the numbers and accidentals of the standard figured bass notation were properly preserved when converting between the three file formats, except for MEI to MusicXML or MEI to **kern, where all figured bass information was lost in both cases. There were also some additional issues with the three aspects of figured bass annotations introduced in Section 6.1. These issues are shown in Table 6-1 and described in more detail below.

[55] Music21 is able to convert file formats between MusicXML, **kern, and MEI, where any of these formats can be imported as a music21 object and exported as any of these formats. However, figured bass information is lost in this process. We used music21 to convert from MEI to MusicXML, since there are no other options.
[56] We referred to the Neue Bach Ausgabe (NBA) critical edition (J. S. Bach, Dürr, and Neumann 1954–2007) for figured bass encodings.
[57] Figured bass encoding instructions for MusicXML: https://musescore.org/en/handbook/figured-bass.
[58] Figured bass encoding instructions for **kern: https://doc.verovio.humdrum.org/humdrum/figured_bass/.
[59] Figured bass encoding instructions for MEI: https://music-encoding.org/guidelines/v4/elements/fb.html.

*Table 6-1: The results of how much figured bass information is preserved when converting between the MusicXML, **kern, and MEI formats, based on the particular results of BWV 33.6. The first column indicates the original format, and subsequent columns indicate target formats. We examined the three figured bass aspects mentioned in Section 6.1, where 1 indicates figures with slashes, 2 indicates continuation lines, and 3 indicates multiple figures over a stationary bass. The first row of each cell indicates the software used for the conversion, based on the experiments by Nápoles López, Vigliensoni, and Fujinaga (2019). "Yes" means the conversion was successful.*

	MusicXML	**kern	MEI
MusicXML		*musicxml2hum* 1: Yes 2: No 3: Yes	*Verovio* 1: No 2: No 3: No
**kern	*hum2xml* 1: No 2: No 3: No		*Verovio* 1: Yes 2: Yes 3: Yes
MEI	*music21* 1: No 2: No 3: No	*mei2hum* 1: No 2: No 3: No	

- *MusicXML to **kern (musicxml2hum[60]):* The continuation line could not be converted, and the resulting **kern file had syntactical errors. Conversions worked for chorales with no continuation line.
- *MusicXML to MEI (Verovio[61]):* Accidentals and slashes were all missing; continuation lines were missing; although figures over a stationary bass were preserved, they all shared the same "tstamp" value, which should be different.

[60] https://github.com/craigsapp/humlib
[61] https://github.com/rism-ch/verovio

- ***kern to MusicXML (hum2xml[62]):* No slashes or continuation lines were converted properly, and figures over a stationary bass were partially lost. The reason is that figured bass is converted as lyrics, rather than the "<figured-bass>" tag MusicXML natively supports for figured bass encodings.
- ***kern to MEI (Verovio[63])*: The conversion was perfect, meaning all the figured bass encodings were properly preserved in MEI.
- *MEI to MusicXML (music21[64]) and MEI to **kern (mei2hum[65])*: All figured bass information was lost.

Regarding all the alternatives that convert MusicXML to the other two symbolic formats, we found that MusicXML to **kern using *musicxml2hum* worked well, except for the continuation lines, and the conversion from **kern to MEI using *Verovio* was perfect. However, direct conversion of figured bass from MusicXML to MEI (using *Verovio*) was problematic, as accidentals, slashes, and continuation lines were not converted. We therefore first converted from MusicXML to **kern, and then manually added the continuation lines to the **kern files using a text editor. The resulting **kern files were then converted to MEI files in the release version of BCFB. Regarding the translated **kern and MEI files, we randomly sampled five chorales for each of these two formats and verified that these files were in good quality.

The FBAs of BCFB serve as a great resource for musicological and MIR study of figured bass, and one of its applications is automatic figured bass annotation, which will be introduced in Section 6.3.

6.3 Automatic figured bass annotation

Now that we have a dataset of figured bass (BCFB), we can attempt to automatically generate figured bass. The generated FBAs use flats and sharps to respectively indicate lowered and raised intervals, and do not contain continuation lines, and they were intended to closely resemble those that Bach might have written himself. This was implemented to predict figured

[62] https://github.com/craigsapp/humextra
[63] https://www.verovio.org/
[64] https://github.com/cuthbertLab/music21
[65] https://github.com/craigsapp/humlib/

bass for those Bach chorales for which no FBAs exist, and the generated FBAs have potential to facilitate other music research (see Section 6.3.5). We used both rule-based and machine learning algorithms to perform this automatic figured bass annotation:[66] The rule-based approach has the potential to model Bach's style of writing figures in ways that are easily human-interpretable, and machine learning has the potential to model patterns in Bach's style that might be difficult to codify into precise, direct rules. We therefore explored the efficacy of both approaches. The data used for this study will be introduced in Section 6.3.1, and both the rule-based and machine learning approaches will be introduced in Section 6.3.2 and Section 6.3.3, respectively. The discussions of the results will be given in Section 6.3.4, and a summary of this research will be given in Section 6.3.5.

6.3.1 Data

We used 120 chorales out of the full 139 chorales in BCFB to train and test our models. We excluded 12 interlude chorales[67] because they are significantly different from the other largely homorhythmic chorales, and we excluded five other chorales[68] that are barely figured. Finally, we excluded BWV 8.06[69] and BWV 161.06 because they feature irregular textures, such as having an obbligato continuo and/or instrumental part.

6.3.2 Rule-based algorithms

6.3.2.1 Initial simple rule-based algorithm

We began by implementing a simple rule-based algorithm that explicitly labels and sorts all the wrapped intervals above the bass in the generated FBAs. First, the music is segmented into a series of note onset slices. A new slice is formed whenever a new note onset occurs in any musical voice, and each slice consists of the vertical set of notes sounding at that moment. Take m. 3.1 of

[66] The installation guideline for the automatic figured bass annotation algorithm is at: https://github.com/juyaolongpaul/harmonic_analysis/blob/master/Github_Page/installation_guide_FB.md.
[67] These chorales have elaborated instrumental interludes between phrases (BWV 24.06, 76.07, 100.06, 105.06, 113.01, 129.05, 167.05, 171.06, 248.09, 248.23, 248.42, and 248.64).
[68] BWV 16.06, 48.07, 149.07, 195.06, and 447.
[69] ".06" in "BWV 8.06" means this chorale is the sixth movement of the "BWV 8" cantata.

Figure 6-2 as an example: Since the pitch classes above the bass "G" are "G", "D", and "B," the FBA generated is 8/5/3.

We then compared the generated FBAs against Bach's original FBAs, and found that the percentage of exact matches was only 3%. This is partly because Bach did not explicitly label all the intervals above the bass in his FBAs; it is often assumed that both he and other Baroque composers employed FBAs that include what are in effect abbreviations that omit obvious intervals (Williams and Ledbetter 2001; Chew and Rastall 2014). For example, consider m. 3.2.5[70] of Figure 6-2: Although the pitch classes "A" and "D" are present in this slice above the bass "F♯", only "D" is explicitly specified[71] by the figure "6"; "3" is not explicitly indicated, but is nonetheless implied by the understood annotation practice of the time.

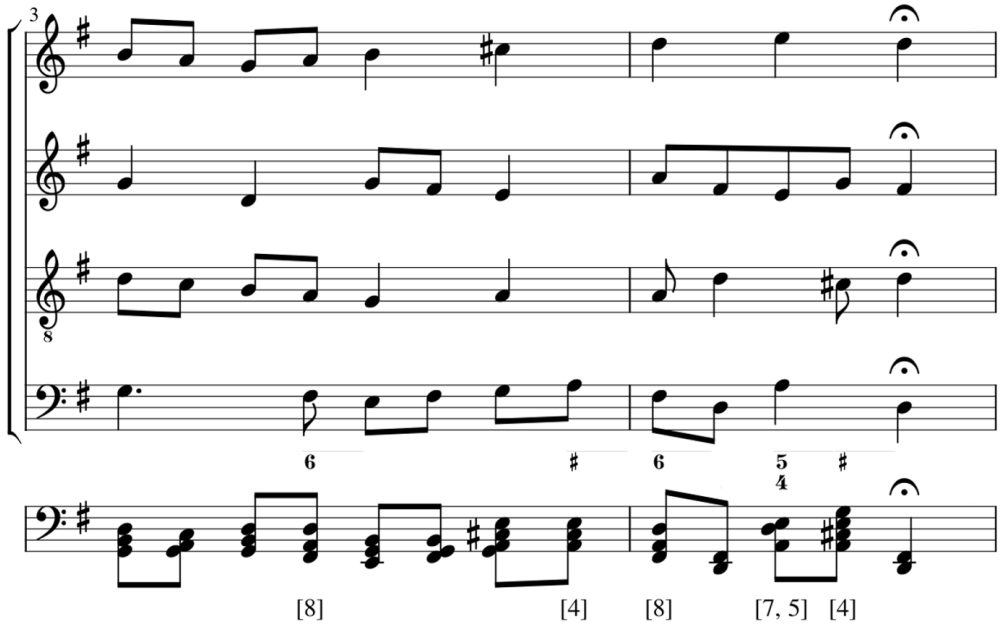


Figure 6-2: Measures 3 and 4 from BWV 117.04 "Sei Lob und Ehr dem höchsten Gut." The original FBAs are annotated underneath the bass voice part. Note that not all slices are necessarily figured, and not all the intervals in a sonority are necessarily specified in FBAs. We artificially added the final bottom staff, which collapses all sonorities into one octave to reveal the pitch-class content more directly. The number of semitones above the bass implied by the original FBAs has also been added underneath this bottom staff. We can also translate this number of semitones back to the original FBAs by examining the actual bass notes in the score and then calculating and labelling the intervals from the bass note.

[70] "m. 3.2.5" means the third measure, on the second and a half beat.
[71] Since "D" forms an interval of a 6th above the bass "F♯".

6.3.2.2 Evaluation metric

To allow for the equivalence in musical content of different figured bass notation conventions, as discussed above, we created an evaluation metric that treats figures that are musically equivalent as notationally equivalent; besides accuracy, this measures the percentage of exact matches between the generated FBAs and Bach's FBAs. The purpose of this metric is to realistically evaluate the generated figured bass when it does not match Bach's figured bass exactly, but does match it materially. The equivalence rules are inspired by Arnold (1931):

- A "3" can be omitted (Figure 6-3a, b, d, and e) unless there is a 4th in the sonority,[72] or unless the 3rd is the resolution of a 4–3 suspension (Figure 6-3c).
- A "5" (Figure 6-3a, c, and d) can be omitted, unless one of the following conditions is true: There is a 6th (Figure 6-3b) in the sonority, the 5th is the resolution of a 6–5 suspension, or the 5th has an accidental (Figure 6-3e).
- An "8" (Figure 6-3c and d) can be omitted, unless one of the following conditions is true: There is a 9th (Figure 6-3b) in the sonority, the 8th is the resolution of a 9–8 suspension, or the 8th has an accidental.
- A "6" can be omitted if the sonority forms a "6/4/3" or a "6/4/2" chord, as shown in Figure 6-3f.

In order to see how the evaluation metric based on these rules operates in practice, consider Figure 6-3a as an example: We can see that "5" and "3" can be omitted, which means "7", "7/3", "7/5", and "7/5/3" are all considered equivalent. Therefore, if the ground truth and the generated figured bass respectively consist of any pairing of "7", "7/3", "7/5", or "7/5/3", then the generated figured bass will be considered correct by the metric.

[72] "Sonority" means the set of pitch classes present in a note onset slice. For example, the added bottom staff of Figure 6-2 shows the sonorities of the four voices for each slice. "4th" means that a wrapped interval of a 4th can be found between the bass and an upper voice, regardless of whether it is labelled in the figured bass.

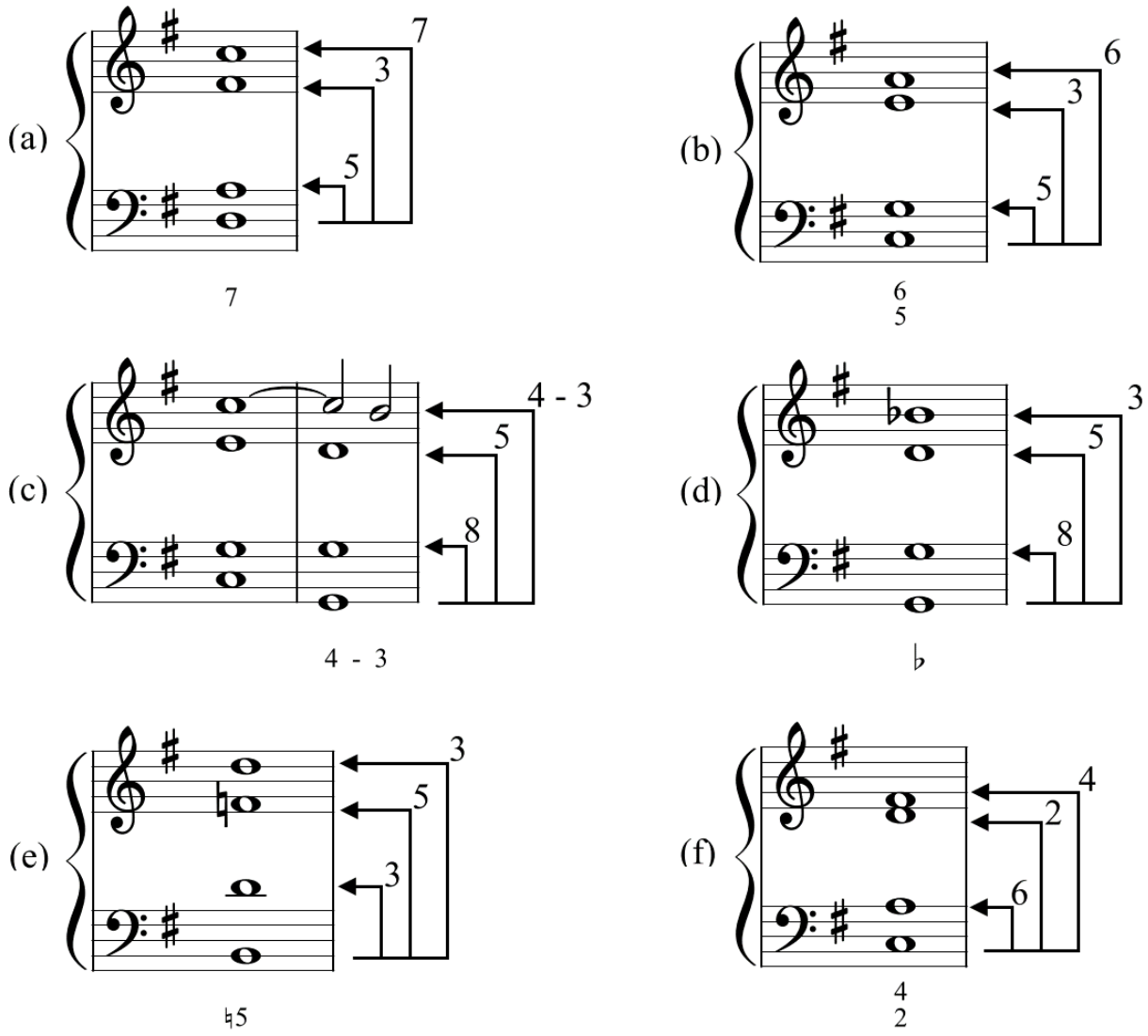


Figure 6-3: Common examples of standard figured bass abbreviations taken into account by the evaluation metric explained in Section 6.3.2.2. In each of the six examples (a)-(f), all the intervals above the bass are shown to the right of the notes connected with arrows, and abbreviated FBAs for the chords are shown below the notes. For example, (a) consists of a dominant 7th chord in root position, and includes notes that are a 3rd, 5th, and 7th above the bass: The figured bass consists of only a "7", with the "3" and "5" omitted, as is often the practice in FBAs.

6.3.2.3 Improved rule-based algorithm

Using the evaluation metrics described in Section 6.3.2.2, the simple ruled-based algorithm described in Section 6.3.2.1 has an effective agreement of 64.5% with Bach's FBAs. We found that when they disagreed, the generated FBAs tend to have more figures than those of Bach (i.e., Bach tends to leave more slices unlabeled than the algorithm). To improve this, we manually

developed additional rules for omitting certain figures, allowing us to predict Bach's style of annotation better.[73] The rules were proposed by observing the generated FBAs and were evaluated against the ground truth FBAs from BCFB. We selected the rules that yielded higher accuracy.

First, we examine each note onset slice and omit the figure for a given note in an upper voice if both of the following two conditions are met: (1) The note is labelled in the previous slice, and (2) the pitch class of the bass in the current slice remains the same as in the previous slice.

Then, we consider slices on fractional beats (e.g., beat 2.5 and 3.5), looking for ornamental notes, which are all approached or departed by step (passing tones, neighbour tones, escape tones, and anticipations). If such a note is in an upper voice, its corresponding number is removed from the FBA; if such a note is in the bass, the slice is left entirely unfigured, with no FBA.

After adding these rules, the model was able to achieve 85.3% agreement with Bach's figures, using the evaluation metrics described in Section 6.3.2.2, a large improvement over the 64.5% agreement achieved with the simple method and equivalency rules.

Although it would have been possible to invest more time in manually analyzing Bach's figured bass to develop still more rules to improve the agreement, none were readily obvious from the perspective of music theory or performance practice, and we wanted to avoid overfitting our rules. So, we turned our attention to machine learning, to see if it could be employed to model Bach's FBA style with equal or better results.

6.3.3 Machine learning algorithms

Unlike the manually-derived rules implemented in the rule-based algorithms in the sections above, in this section we explore machine learning methods to automatically learn to perform automatic figured bass annotation in its training—something that has never been explored in the literature to the best of our knowledge. In Section 6.3.3.1 and Section 6.3.3.2, we will introduce the input features of the machine learning algorithms. The experimental setup and the results of

[73] These rules were developed through an iterative process of improvement on the whole dataset; a reserved test set was not used to measure the performance of these rules, so there is the possibility that the development of the rules may have potentially involved some overfitting to this particular music.

the machine learning algorithms will be introduced in Section 6.3.3.3 and Section 6.3.3.4, respectively.

6.3.3.1 Transformation from figured bass to interval classes

Recall that figured bass indicates intervals above the bass note. Thus, for each slice, one can convert the figures to an interval-class vector. As an important note, we are using the term "interval class" here to refer to *ordered* interval class. Since we wish to calculate the intervallic relationship *from* the bass *to* an upper voice (in that order), we wish to distinguish between intervals and their inversional equivalents (e.g., minor sevenths are distinguished from major seconds). Thus, ordered interval classes range from 0 to 11, while unordered interval classes range from 0 to 6 only.

An (ordered) interval class, which is conceptually similar to a pitch class, is a set of intervals wrapped by octaves. For example, an interval class of a major second includes a major ninth and all other octave expansions of a major second. As with a pitch-class vector, an interval-class vector contains 12 elements, representing intervals in semitone increments. In our case, each FBA is converted to an interval-class vector that includes all the notes above the bass that are sounding in the current slice. In cases where the figured bass does not specify the exact interval in semitones, such as the "6", which could be either a major sixth or a minor sixth, we rely on the score to determine the exact interval using heuristics-based processing.[74] For example, if the bass is C and there is "6" above it, the implied interval will be a major sixth (nine semitones above the bass) if the key signature of the score specifies no accidentals for A, and a minor sixth (eight semitones above the bass) if the key signature has a flattened A. We similarly rely on the score to later convert interval-class vectors back to figures: For example, an interval of three semitones can be interpreted as either a minor third (figure "♭3") or an augmented second (figure "♯2"). We can decide which is appropriate by considering the pitch spelling in the original score.[75] This interval

[74] On rare cases, the interval indicated by the figure cannot be found in the score (see the detailed discussions in Section 6.4.2.4), and we discarded such figures in automatic figured bass annotation.
[75] For example, to distinguish figures "2" and "9" when unwrapping interval-class vectors, we need to find the actual note above the bass and compare its pitch to the pitch of the bass. If they are one octave apart, the generated figure will be "9", and "2" otherwise.

class representation of the figured bass is used for both input and output of the machine learning algorithms.

6.3.3.2 Input features and machine learning algorithms

The three feature vectors used as input to the machine learning algorithms are: (1) Interval class vectors (see Section 6.3.3.1); (2) onsets, which specify all the notes in the musical surface (above the bass) that have onsets within the slice, as opposed to being held from a previous slice; and (3) metrical context, which specifies whether a slice occurs on the downbeat of a measure, on another whole beat (e.g., beat 2, 3, or 4 in 4/4), or on a fractional beat (e.g., beat 3.5). These binary vectors are specified for each slice. The following example demonstrates each of these feature vectors for m. 4.2.5 of Figure 6-2 (the bit-length of each feature is indicated in parentheses):

- *Interval classes* (12): The bass note is "A" (held), with pitch classes of "C♯", "G", and "E" (held) above it, which are respectively four, ten, and seven semitones away from it. The feature vector is thus: [0,0,0,0,1,0,0,1,0,0,1,0].[76]
- *Onsets* (12): "C♯" and "G", which are respectively four and ten semitones above the bass, are the pitch classes with onsets on this slice, so the feature vector will be [0,0,0,0,1,0,0,0,0,0,1,0].
- *Metrical context* (3)*:* Because the slice is on beat 2.5 of a 4/4 measure, the feature vector will be [0,0,1] (i.e., it is a fractional beat).

Each slice, therefore, is represented as a 27-dimensional (12+12+3=27) binary vector. To provide a context for each slice, the machine learning algorithms are also provided with the two 27-dimensional vectors for the previous and following slices (zero-padded for the first and the last slices). Thus, the total length of the input vector for each slice is 81 (27×3=81).

We experimented with two machine learning algorithms: Decision Trees (DT)[77] and Deep Neural Networks (DNN),[78] as the former is able to learn the rules from data explicitly, and the latter can be powerful in learning the mapping between inputs and outputs (Seide, Li, and Yu 2011).

[76] The first dimension indicates a unison (or collapsed octaves).
[77] We used the "`DecisionTreeClassifier`" function from the "`scikit-learn`" library, under default settings. This function is an optimized version of CART (Classification And Regression Tree).
[78] We used a feedforward network with three hidden layers, each with 300 hidden units. Adaptive Moment Estimation was used as an optimizer, with a binary cross-entropy-based loss function.

Both algorithms used the input features specified above. Their output each consisted of a 12-dimensional binary interval class vector specifying the number of semitones above the bass,[79] as discussed in Section 6.3.3.1. An illustration of the input and output vectors of the DNN model is shown in Figure 6-4.

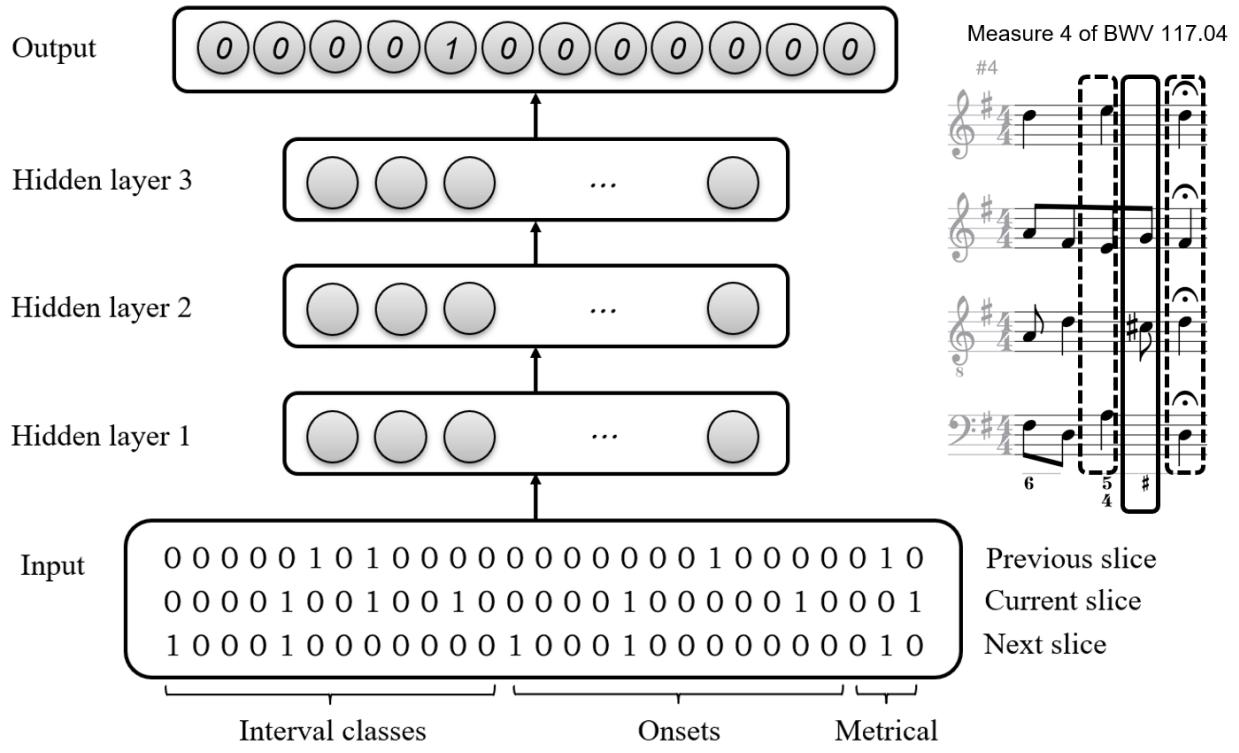


Figure 6-4: Illustration of our machine learning approach for automatic figured bass annotation, using the DNN architecture as an example (the decision tree architecture uses the same input and output formats). The input and output vectors are illustrated using the example of m. 4.2.5 of BWV 117.04, shown on the right. Note that the slice with the solid line rectangle is the current slice, and the directly preceding and following slices (with dashed line rectangles) are concatenated as context in the input vector introduced in Section 6.3.3.2.

[79] For example, the output vector for m. 4.2.5 of Figure 6-2 will be [0,0,0,0,1,0,0,0,0,0,0,0], which indicates a single note with an interval four semitones above the bass (an A in this case), which suggests a C# in this particular tonal context. This corresponds to an FBA of "♯".

6.3.3.3 Experimental setup

Ten-fold cross-validation was used for evaluation. For the DNN experiments, we divided the data for each fold into training (80%), validation (10%),[80] and testing (10%) folds. For the DT experiments, the data was divided into training (90%, the union of the DNN training and validation sets) and testing (10%, matching the DNN test sets) partitions.

6.3.3.4 Results

The Decision Trees and Deep Neural Networks respectively achieved classification accuracies of 84.3±0.5% and 85.9±0.6% on BCFB.[81] These accuracies are calculated based on the evaluation metrics proposed in Section 6.3.2.2.

6.3.4 Discussion

It is useful to examine the types of errors that our model made, in order to better understand its performance and how it can be improved. We will focus this discussion on the two musical examples shown in Figure 6-5, as they are representative of the kinds of errors our model made, based on an informal sampling of the results. One common error made by our model was to miss figures that indicate the resolution of a suspension, such as the 9–8 shown in Figure 6-5(a), m. 8.4. This may be because the features we used did not contain sufficient voice-leading information to detect such suspensions.

Two more types of disagreement between our model and Bach's figures are shown in Figure 6-5(b). At m. 2.3 our model generated "♯", but the ground truth had no label. In fact, the generated "♯" is technically correct, as the D is explicitly sharpened in the soprano. Turning to m. 3.2, our model's prediction included a "♯7," unlike the ground truth. Perhaps this suggests that Bach might have considered the corresponding "D♯" to be a passing tone? Or perhaps the D♯ was understood as a "diatonic" note in this Dorian chorale tune? At any rate, the "♯7" in the generated

[80] The hyperparameters of DNN were tuned using the validation set. Since ten-fold cross validation is used, this validation set therefore covers different parts of the data across all folds. I chose the hyperparameters that resulted in the highest mean value of classification accuracy in this validation set across all folds. The validation set was also used for the selection of the DNN model in each fold with the lowest validation error during the training for that fold.

[81] Uncertainty values show standard error across cross-validation folds.

figures should not necessarily be considered wrong. Both these figures are in fact theoretically acceptable answers, although they may not necessarily be representative of Bach's individual stylistic approach to writing figured bass.

Such differences between the ground truth and the predicted figures are intriguing, as they hint at contrapuntally or harmonically meaningful information present in Bach's figures that is not explicit in the four vocal lines. Or perhaps they are of negligible meaning? It is impossible to know with the information we have now, but future comparisons with models trained not just on Bach but on the figures of many Baroque composers could potentially reveal fascinating insights on figured bass style.

We also observed interesting variability in the types of figures Bach used under seemingly similar musical contexts. Three examples are shown in Figure 6-5:

- *Accidentals:* Figure 6-5(b), Bach did not label the first "♯" at m.2.3, but did label the second one at m. 3.3.
- *Suspensions:* Bach sometimes labelled suspensions (e.g., m. 8.4 of Figure 6-5(a)), and sometimes omitted them (e.g., m. 1.4 of BWV 194.06 [not shown]).
- *The same chord:* Bach sometimes labelled a 6/4/2 chord as a 4/2 chord (e.g., m. 2.1 of Figure 6-5(b)), and sometimes as a 6/4/2 chord (e.g., m. 10.4 of BWV 13.06 [not shown]).

So, one cannot necessarily reasonably expect 100% agreement to be achievable with Bach's specific annotations, given such variabilities. Bach, like everyone, was sometimes inconsistent with himself, which imposes an artificial performance ceiling on attempts to model him (Flexer and Lallai 2019). In any case, these types of variabilities can be of great interest to music theorists and musicologists, and offer significant potential for future research.

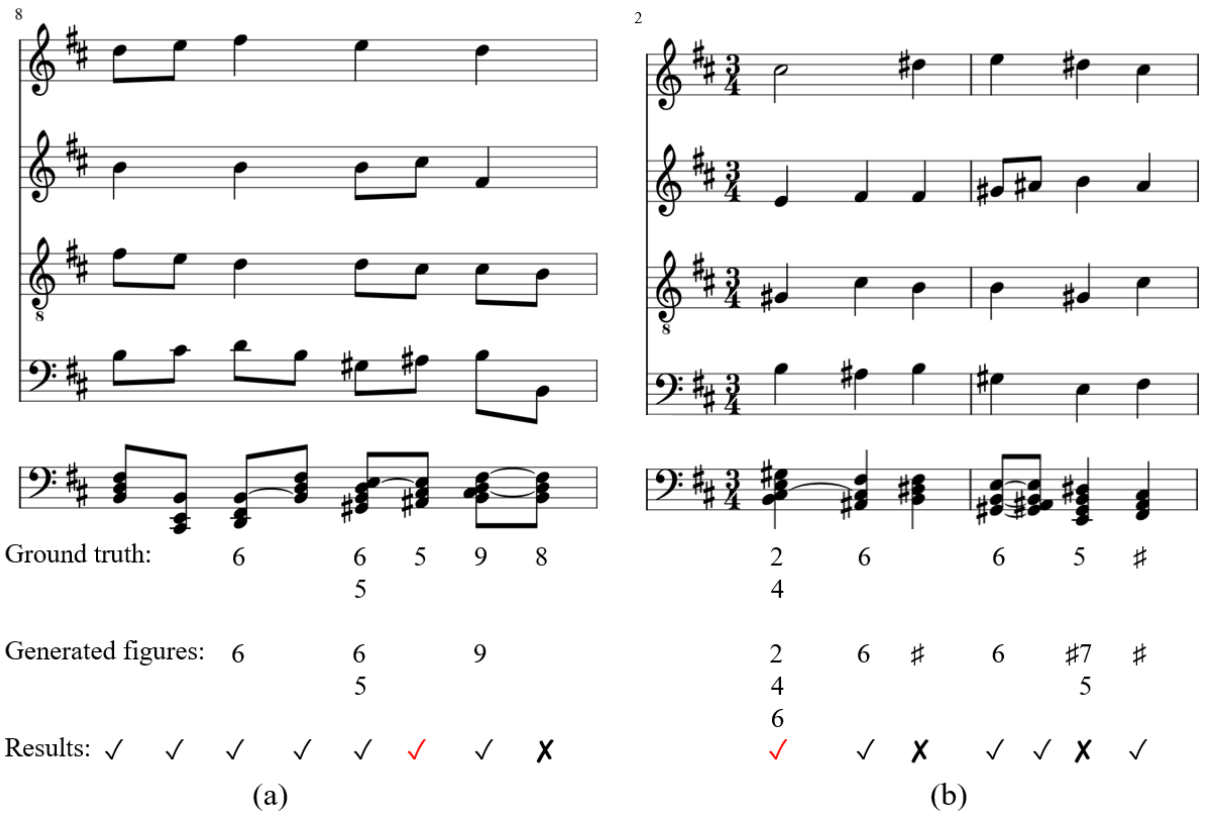


Figure 6-5: An illustration of figured bass generated by our best-performing model for measure 8 of BWV 108.06 "Es ist euch gut, daß ich hingehe", and measures 2 and 3 of BWV 145.05 "Ich lebe, mein Herze, zu deinem Ergötzen", which are labelled (a) and (b) here, respectively. We artificially added the fifth (bottom) staff, which collapses all sonorities into one octave to reveal the pitch-class content more directly. As discussed in Section 6.3.3.1, our model predicts interval classes, and the figured bass is generated based on the intervals between the bass note and each predicted interval class. The agreement of each prediction with Bach's FBAs are shown as well: "✓" means that the generated figured bass exactly matches Bach's FBAs (the ground truth), "✓" in red means they are considered correct by our evaluation metric that treats musically equivalent figures as equivalent (see in Section 6.3.2.2). An example of the latter can be found at m. 2.1 of (b) where the generated figures can be reduced to "2/4" from "2/4/6" (since the "6" can be omitted, as discussed in Section 6.3.2.2). "✗" means the generated figures are considered to be errors in our evaluations.

6.3.5 Summary

This section showed how BCFB could be used as the basis for developing and evaluating both rule-based and machine learning models for predicting figured bass; our models respectively achieved classification accuracies of 85.3% and 85.9% ± 0.6% on BCFB, based on the modified

accuracy metric proposed in Section 6.3.2.2. A potential reason the machine learning models did not outperform the rule-based model may be the relatively small size of BCFB.

Figured bass automatically generated by our models could help performers improvise *basso continuo* accompaniment for the remaining unfigured Bach chorales, or inform the design of pedagogical software for teaching Baroque theory or composition. Of particular interest, figured bass can potentially benefit automatic chord labelling research. Existing methods tend to either identify chords directly from the music (Chen and Su 2019; Masada and Bunescu 2019; Granroth-Wilding 2013), or identify and remove non-chord tones from the score and then generate chord labels from the remaining chord tones (Condit-Schultz, Ju, and Fujinaga 2018; Ju et al. 2019). A new approach would be first to generate figured bass automatically from the music and then convert the figures to chord labels; this would allow a chord classifier to take advantage of knowledge implicitly learned from Bach's ground truth FBAs by a figured bass annotator during its training.

There are several refinements that could potentially improve the quality of the figured bass our approaches generate. The first is to add voice-leading information (how one voice moves horizontally), which may reduce some of the errors discussed in Section 6.3.4. The second would be to improve our rule-based model (e.g., by analyzing automatically trained decision trees), which in turn could provide further insights into Bach's approach to figuring bass, and perhaps provide musicological insight on how his methods changed over time or by the context of performance.

Another potential extension to this research would be to incorporate FBAs from other pieces by Bach, such as his chamber music, or pieces by other Baroque composers, which are usually figured throughout the Baroque period. Once we have a variety of figured bass datasets for different genres and composers, we may then be able to train models that generalize better. Also, by comparing Bach's FBAs to FBAs by other composers, we may gain meaningful insights into Bach's unique figured bass style and discover a sense of the degree of stylistic variability with which composers approached figured bass.

One of the goals of this dissertation is to be able to generate multiple parallel chord label annotations based on music such as those in BCFB. We therefore propose a series of four different rule-based algorithms that generate chord labels automatically based on both the figured bass annotations and the musical surface. These algorithms and the Bach Chorales Multiple Chord Labels dataset they produce will be introduced in Section 6.4 and Section 6.5, respectively.

6.4 Automatic chord labelling: A figured bass approach

Automatic chord labelling can be challenging, largely because the identification of chords directly from the musical surface can be ambiguous. Figured bass can potentially offer indications of harmonic rhythm and non-chord tones, thereby reducing this ambiguity. In this section, I will propose a series of four rule-based algorithms that automatically generate chord labels for homorhythmic Baroque chorales based on both figured bass annotations and the musical surface. These are applied to the existing Bach Chorales Figured Bass dataset (Ju, Margot, McKay, Dahn, et al. 2020) presented in Section 6.2, which consists of 139 chorales composed by Johann Sebastian Bach, and includes both the original music and figured bass annotations.

6.4.1 Introduction

As a way of approaching automatic chord labelling, figured bass annotations (FBAs, see detailed introductions at Section 6.1) are a type of music notation commonly used in Baroque music that comprises numerals and other symbols to indicate intervals above the bass line (or "continuo"). These are considered one of the earliest historical ways to imply chord-like labels (Bach [1753] 1949; Williams and Ledbetter 2001). We propose innovative, rule-based algorithms for automatic chord labelling that consider both the musical surface and figured bass. Compared to existing methods that only considered the musical surface (Tsui 2002; Kröger et al. 2008; Masada and Bunescu 2017; Condit-Schultz, Ju, and Fujinaga 2018; Chen and Su 2018; 2019; Micchi, Gotham, and Giraud 2020; Koops et al. 2020), the advantages of this approach are:

- Figured bass offers some indications of harmonic rhythm[82] and non-chord tones (Holtmeier 2007; Inman 2018),[83] thereby potentially reducing the level of harmonic ambiguity. Figure 6-6 provides an example of how it can be ambiguous to label certain chords without FBAs: Two chord labelling analyses based only on the musical surface, resulting in disagreement at six spots in just two measures of music. Fortunately, FBAs may help to resolve such ambiguity. For example, in m. 5.2.5 the FBA indicates a possible chord change,[84] which suggests the passing D should be interpreted as a chord tone, thus forming a new "Em7" chord. In the cases of m. 5.3.5 and m. 5.4.5, both spots are left unfigured, suggesting no chord change, which means that the passing notes might be interpreted as non-chord tones, so the chord labels remain unchanged from the preceding slices ("F" and "Bo", respectively).
- Since FBAs are often attributed directly to composers, as opposed to copyists or editors, the chord labels they imply may offer meaningful insights into a composer's unique compositional style.[85] This applies especially to intentions relating to counterpoint and harmony, which are of interest both to the generation of authoritative ground truth for automated systems and to music-theoretical and musicological study.

[82] Harmonic rhythm means the rate at which chords change in the music.
[83] Non-chord tones that are part of suspensions are indicated in the FBAs by voice-leading motions (e.g., 9–8). Other types of NCTs, such as passing tones or neighbor tones, are implied by the absence of the corresponding notes in FBAs.
[84] In this paper, we consider the harmony theory of 18-century (Bach [1753] 1949) that each slice with figures represents an individual chord.
[85] Although figured bass is primarily a notation for performance, rather than a strict prescription of harmony, it nonetheless provides a useful description of harmony acknowledged by others (Bach [1753] 1949; David, Mendel, and Wolff 1999).

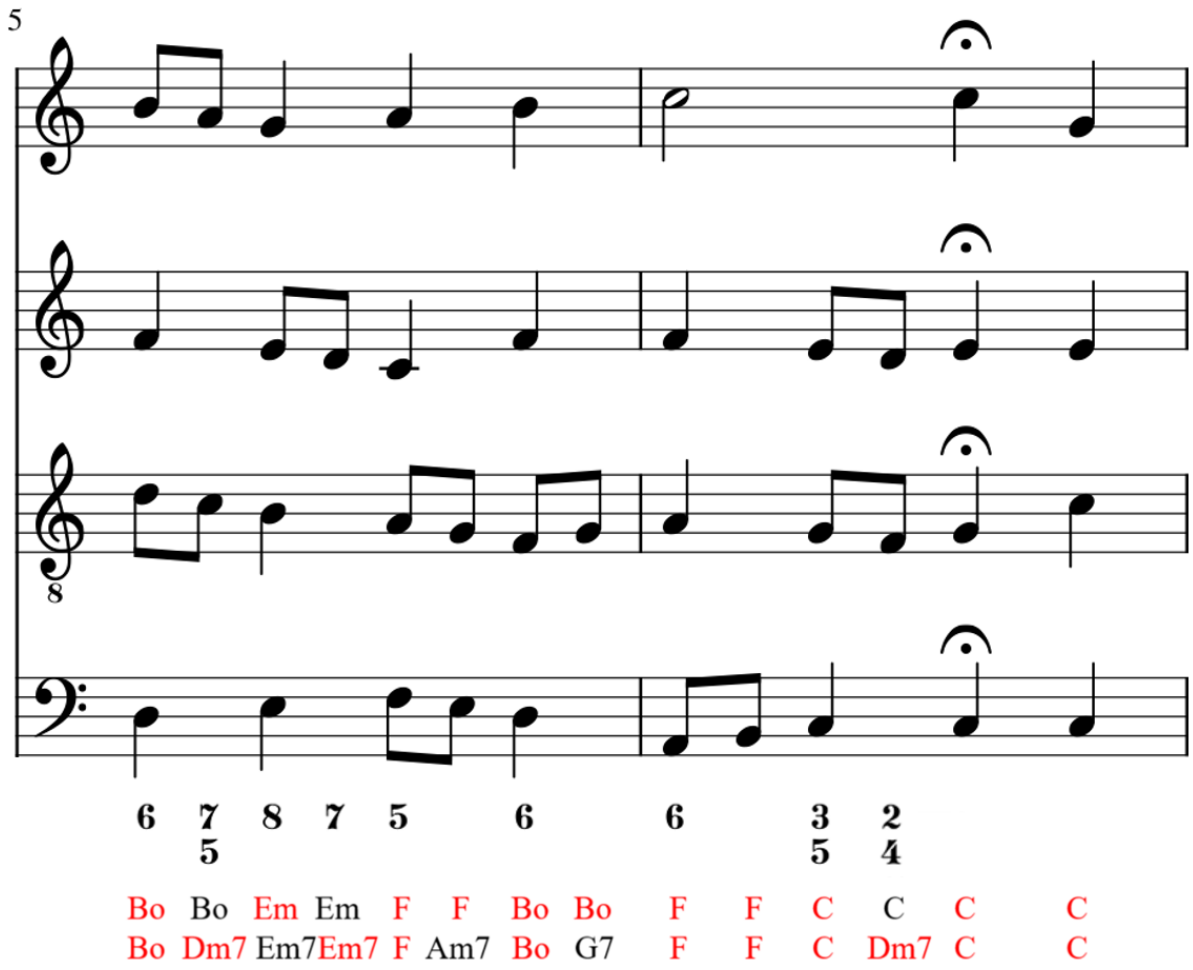


Figure 6-6: Measures 5 and 6 of BWV 248.05 "Klagt, Kinder, klagt es aller Welt." Two rows of chord labels represent two possible analyses based only on the musical surface. One can see that their annotations did not always agree, which suggests a degree of harmonic ambiguity. Figured bass can help resolve such disagreements. Chord labels supported by the figured bass are marked in red.

6.4.2 Methodology

We propose a series of rules that can be applied to generate chord labels from both FBAs and the musical surface. These rules are based on treatises that discussed figured bass and chords (Arnold 1931; Bach [1753] 1949), and on consultation with expert music theorists.[86] For ease of understanding and to permit empirical comparisons, we have divided the associated processing

[86] Samuel Howes and Sylvain Margot, who are Ph.D. music theory students from McGill University.

steps into four algorithms (A, B, C, and D), each of which successively incorporates all of the processing in the previous algorithm and also adds additional new rules.

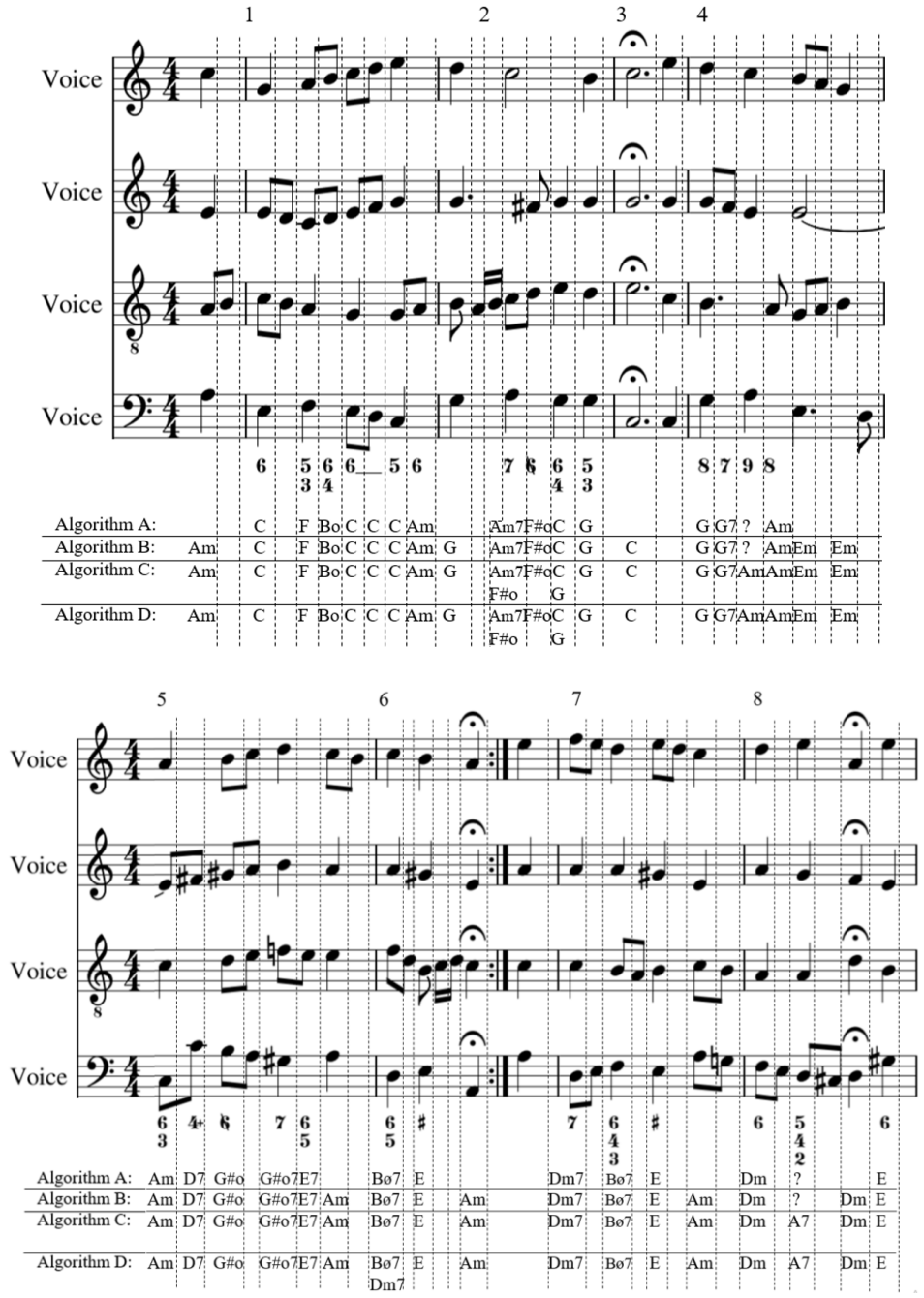


Figure 6-7: The first measures of BWV 33.06 "Allein zu dir, Herr Jesu Christ" from our Bach Chorale Figured Bass (BCFB) dataset. FBAs and chord labels are shown below the bass line. The vertical dashed lines divide the music into a series of note onset slices, which are formed whenever a new note onset occurs in any voice; each slice consists of the vertical set of pitch classes sounding at that moment. The results produced by each of our four chord labelling algorithms (see Section 6.4.2) are indicated below the music, separated by horizontal lines.

As an initial pre-processing step, the music is segmented into a series of note onset slices, which are formed whenever a new note onset occurs in any voice; each slice consists of the vertical set of pitch classes sounding at that moment, along with the figured bass annotation for the slice from the original score, as shown in Figure 6-7. Then, for each slice, each of the four algorithms outputs one or more candidate root pitch classes and one or more of nine candidate chord qualities: Major, minor, diminished, and augmented triads, as well as major, minor, dominant, half-diminished, and fully diminished 7th chords.[87] Figure 6-7 demonstrates a sample output for each of the algorithms. The open-source code implementing each of these algorithms can be found at: https://github.com/juyaolongpaul/harmonic_analysis/blob/master/Github_Page/installation_guide_FB_chord.md.

6.4.2.1 Algorithm A

This is the baseline algorithm, which generates chord labels based only on the bass notes and the FBAs (unlike algorithms B, C, and D, which also consider all voices of the musical surface). For each figured slice, Algorithm A will note all the pitch classes (PCs) implied[88] by the FBA as chord tones above the bass note. If the quality of the chord outlined by these notes has one of the nine candidate qualities specified above, then the slice is labelled with this chord. Otherwise, the slice will be labelled with "?". There is also additional logic regarding which figures are implied from the FBA, based on the contemporary rules by Johann Heinichen and George Telemann, and summarized in Arnold (1931, 263, 311).[89] The output of Algorithm A is always a single chord label (root pitch class and quality, or "?") for each slice with a figure, and no label for slices without figures.

[87] We limit the output to just these qualities for the sake of simplicity, and because they are commonly discussed in music theory associated with the Baroque period.
[88] Take m. 5.3 of Figure 6-7 as an example. Here the figure is "7", which indicates a root position seventh chord above the bass "G♯." We know that by definition a seventh chord consists of intervals of a 3rd, 5th, and 7th above the root, so next we can look at the key signature to find the diatonic pitch class set, which in this case is [C, D, E, F, G, A, B]. Therefore, a 3rd, 5th, and 7th above the bass respectively indicate pitch classes of "B", "D", "F", which along with the bass "G♯" represent a "G♯o7" chord.
[89] For example, m. 5.1.5 of Figure 6-7 the "4+" also implies "6" and "2" over the bass, forming a "D7" chord.

6.4.2.2 Algorithm B

It was common shorthand for Baroque composers to omit figures corresponding to root position major or minor triads (e.g., mm. 2.1, 3.1, and 4.3 of Figure 6-7). This, of course, does not imply that such slices are not harmonically important or that accompanists should ignore them. Algorithm B, a superset of Algorithm A, therefore, introduces new processing that considers the full musical surface (as well as the FBAs) to label all unfigured slices consisting of root position triads. Overall, Algorithms A and B focus on converting FBAs to chord labels according to 18th-century treatises, without imposing newer theories of chord labelling (e.g., suspensions or non-chord tones in the bass).

6.4.2.3 Algorithm C

Departing from Algorithm A and Algorithm B, Algorithm C begins to incorporate newer approaches to chord labelling. Suspensions are an indispensable part of modern chord labelling practice, so Algorithm C (a superset of Algorithm B) identifies suspensions based on both the FBAs and the musical surface. There can, however, be two ways of legitimately labelling suspensions with chords. For example, in the case of "7–6" suspensions, "6–5" suspensions, and cadential "6/4" suspensions[90] (established by either "4–3" or "6/4–5/3" suspensions), suspended notes can be either treated as chord tones, resulting in, respectively, a seventh chord (e.g., "Am7" at m. 2.2 of Figure 6-7), a sixth chord, or a 6/4 chord (e.g., "C" at m. 2.3 of Figure 6-7), or they can be treated as non-chord tones, in which case they should adopt the chord label from the slice to which the suspension is resolved (e.g., "F♯o" at m. 2.2 and "G" at m. 2.3 of Figure 6-7). Algorithm C outputs up to two possible chord labels in such cases, in order to reflect both theoretically viable options. Other slices not involving such suspensions are each annotated with a single chord label, according to Algorithms A and B. Additionally, if a slice is figured (e.g., the "5/4/2" slice at m. 8.2 of Figure 6-7) but the chord quality (the nine candidate qualities specified in Section 6.4.2) is not one of the recognized types (slices where Algorithm A and Algorithm B will output "?"), then the algorithm adopts the chord label from the subsequent[91] slice (Bach [1753],

[90] Here, an en dash (–) represents the voice leading motion and a slash (/) separates multiple figures within a single FBA.
[91] In such cases, the chord is labelled based on the musical surface only.

1949, p. 196 §75 and §76; p. 425 §1 to §3). This can be seen, for example, in the "A7" chord label[92] at m. 8.2 of Figure 6-7.

6.4.2.4 Algorithm D

It can happen that a PC explicitly specified by the figured bass is not found in the musical surface, such as in m. 6.1 of Figure 6-7, where the bass is "D" and the figured bass suggests a "6/5" chord ("Bø7"), but the musical surface suggests a seventh chord ("Dm7"), because the "B" explicitly specified by the figure "6" is not in the surface. Such discrepancies are valuable to track, not only because they identify two possible chord labels, but also because they demonstrate a situation where the figured bass and the musical surface appear to disagree, which may provide new insights into musicology and music theory. We therefore implemented Algorithm D, which is a superset of Algorithm C, that identifies such discrepancies and adds the corresponding alternative chord label for slices where the figured bass and musical surface disagree. Note that Algorithm D does not track the reverse case; that is, no alternate chord label is generated when a pitch in the surface is not indicated in the figure (e.g., in m 1.4.5 of Figure 6-7, the "G" that would correspond to a 7th of "A" is in the surface, but is not indicated in the figure), since we consider pitches in the surface but not indicated in the figures to be non-chord tones.

6.5 Building the Bach Chorales Multiple Chord Labels (BCMCL) dataset

Here, we introduce the new Bach Chorales Multiple Chord Labels (BCMCL) dataset, which serves as the data to explore multi-label learning and label distribution learning, and this work will be introduced in Chapter 7. This dataset includes 120 of the 139 BCFB chorales (see Section 6.3.1), now annotated with the chord labels output by each of our four algorithms described in Section 6.4.2. The BCMCL dataset is released and publicly available at: https://github.com/juyaolongpaul/BCMCL/releases/tag/v1.0. The chord labels are saved in two ways: (1) MusicXML files, where the chord labels are added underneath the FBAs of the BCFB

[92] This aligns with the figured bass treatise by Heinichen (Arnold 1931, 261), where a "5/4/2" figure is considered as "the first inversion of a Seventh with retarded bass."

dataset and can be found in the folders of the repository for each algorithm (each MusicXML file holds only the chord labels corresponding to the single algorithm for the folder); (2) text files, where the results of each of the four algorithms for all the 120 chorales are saved in a separate text file. Please refer to the README of the repository for examples (https://github.com/juyaolongpaul/BCMCL/blob/master/README.md), as well as a detailed introduction to the BCMCL chord labels.

The parallel tracks of chord labels produced by our four algorithms allow users to access the kinds of labels most relevant to their own research; for example, those conducting historical research may be interested in the Algorithm A or Algorithm B labels, which do not impose any non-contemporary harmonic models, while those working on training automatic chord classifiers based on modern harmonic analysis may be more interested in the Algorithm C or D labels. Comparing the differences between the label tracks may also be of musicological or music-theoretical interest.

6.5.1 Dataset statistics

Table 6-2 summarizes certain statistics on BCMCL, based on the chord labels produced by Algorithm D.[93] The distributions of chord types and qualities are shown in Figure 6-8. Note that the top 20 chord types (identified by the combination of chord root and quality) represent 75.1% of all chords, and these are mostly major or minor triads. Seventh chords are less common, with their most frequent qualities being dominant seventh (7.5%) and minor seventh (5.8%). Augmented triads, major seventh, half-diminished, and fully diminished seventh chords are relatively rare in BCMCL.

[93] We chose to show the statistics of Algorithm D because they demonstrate useful information on the number of slices with two chord interpretations, and on the number of slices with figured bass and surface discrepancies. Slices left unlabelled by Algorithm D (e.g., m. 3.4 of Figure 6-7) are assigned label(s) from the previous slice (e.g., "C" will be the chord label for m. 3.4) for the purposes of Table 6-2, Figure 6-8, and Figure 6-9. Note that this practice was not applied when generating the labels that were actually included in the BCMCL data, so there is therefore a difference between the values listed here and what one sees in the dataset annotations themselves.

Table 6-2: The number of chorales, note onset slices, candidate chord qualities, chord types (identified by the combination of chord root and quality), and chord labels (including all labels for all slices produced by Algorithm D) in the BCMCL dataset. Total slice counts and percentages (divided by the number of note onset slices) are also provided for slices with suspensions (resolutions not included), slices with two possible chord labels, and discrepancies between the figured bass and musical surface (counting asymmetrically, as described in Section 6.4.2.4).

Category	**Number**
Chorales	120
Note onset slices	9,617
Candidate chord qualities	9
Chord types	109
Chord labels (from Algorithm D)	10,092
Slices with suspensions	312 (3.24%)
Slices with two chord interpretations	471 (4.90%)
Slices with figured bass and surface discrepancies	276 (2.87%)

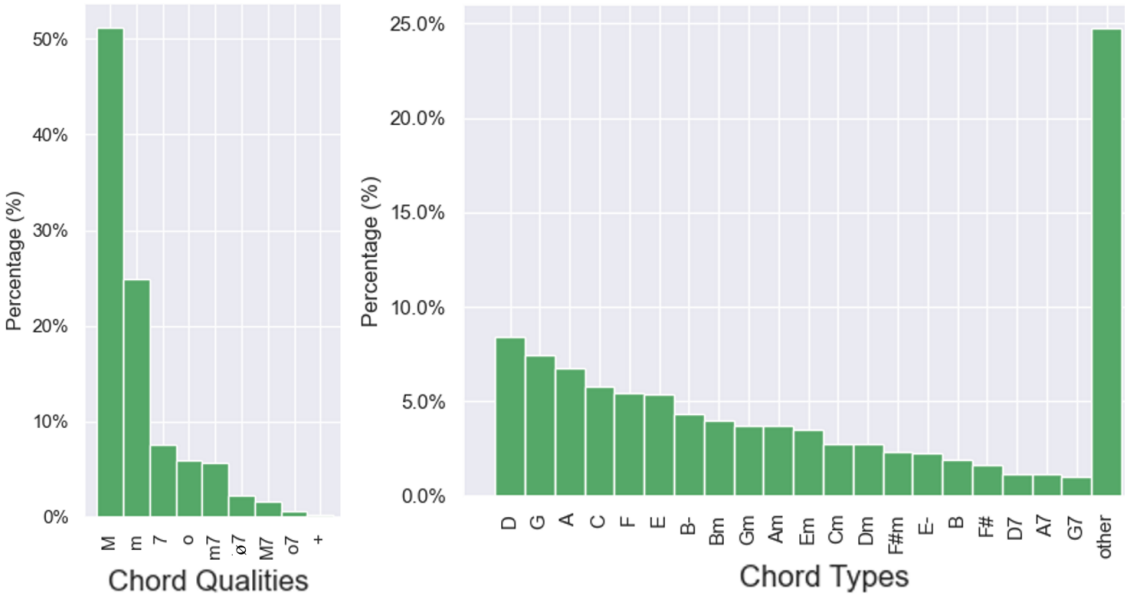


Figure 6-8: Distributions of chord qualities (left) and chord types (right) for all 10,092 chords in BCMCL.

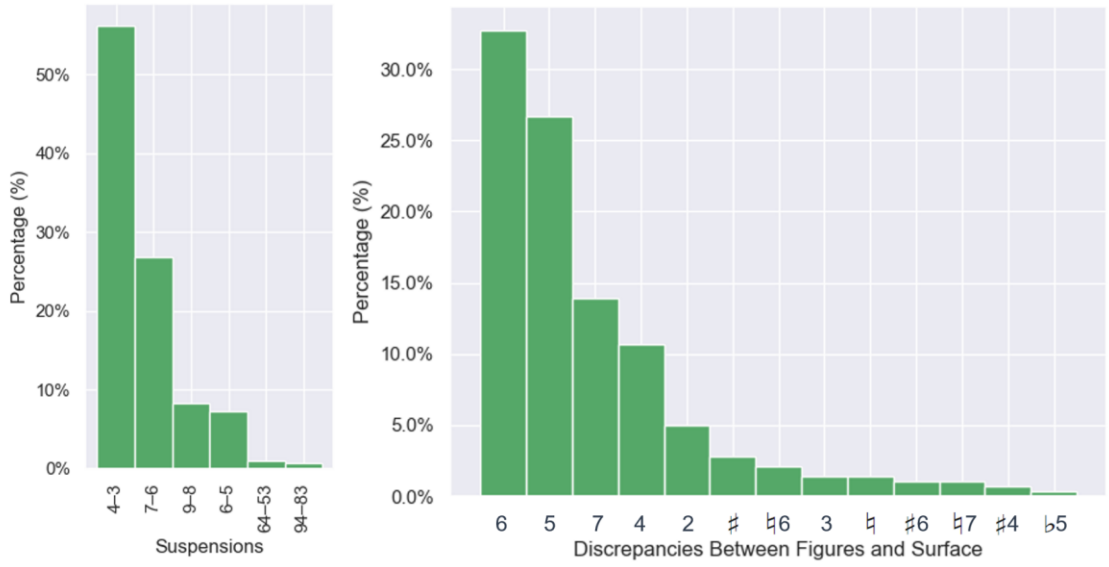


Figure 6-9: Distributions of suspensions (left), and discrepancies between the figured bass and surface (right). In the latter graph, each column corresponds to an interval found in the figured bass but absent in the surface. The figure "3" is sometimes omitted from the figured bass: In such cases, "♯" and "♮" mean raised third and natural third, respectively.

We can observe from the distributions of suspensions (left of Figure 6-9) that the majority of suspensions are of the "4–3" type, followed by "7–6", "9–8", and "6–5" suspensions which are less frequent, but still happen regularly. Double suspensions ("6/4–5/3" and "9/4–8/3") exist but are rare in BCMCL. Also, looking at the discrepancies between the figured bass and the surface (right of Figure 6-9), the pitch classes indicated by the figures "6" and "5" are particularly likely to be absent in the musical surface.

6.5.2 Summary

In Section 6.5, we presented the Bach Chorales Multiple Chord Labels (BCMCL) dataset, which we hope will facilitate future research revealing insights into Bach's unique compositional style, especially plausible possible ideas about his thoughts on counterpoint and harmony. The four separate tracks of chord labels may provide an interesting comparative resource. The statistics calculated on BCMCL may also be of interest to musicological research on Bach's chorales.

Furthermore, BCMCL may be useful in automatic chord labelling research. Offering multiple chord labels per slice, when appropriate, facilitates multi-label learning and label

distribution learning, an understudied but essential aspect of music information retrieval research (see Chapter 7). Even for systems that can only output single classifications per slice, datasets such as this, which specify multiple chords per slice, also serve as a basis for exploring alternate evaluation metrics for automatic chord labellers in general. Despite its usefulness, the chord label annotations of BCMCL were entirely generated heuristically and had no systematic manual or external validation performed on them. Further examinations may be required before considering them as expert ground truth annotations.

There are several particularly intriguing directions for future research. One is to more deeply study the context of the discrepancies between the figured bass and the musical surface; such studies may yield potential insight on how and why Bach figured his chorales. Also, our algorithm currently works only with homorhythmic music, so expanding it to work on music with other textures, especially homophony, would certainly be valuable. Our long-term goal is to facilitate the usage of figured bass and chord annotations by creating a search engine and interface that would make them both searchable via a single interface.

6.6 Conclusion

In this chapter, we first introduced the new Bach Chorales Figured Bass (BCFB) dataset, which consisted of 139 chorales composed by Johann Sebastian Bach. These chorales comprise an integral part of larger choral works composed by Bach: The cantatas, passions, motets, and the Christmas Oratorio. We chose this repertoire due to its key role in modern music pedagogy and its general historical importance. To facilitate the future creation of more figured bass datasets, we included our methodology for digitizing FBAs in an efficient and effective way (Section 6.2).

We then presented a comparative study of automatically generated figured bass annotations of BCFB, using both rule-based and machine learning approaches (Section 6.3). The results were discussed with reference to specific musical examples (Section 6.3.4). We highlighted possible applications of figured bass annotation (Section 6.3.5), and implemented one in connection with converting figured bass to chord labels, where we introduced four rule-based algorithms that automatically generated chord labels based on both figured bass annotations and the musical surface for Baroque homorhythmic chorales, and applied them to the BCFB dataset (Section 6.4).

Finally, we presented the resulting chord label annotations of the four rule-based algorithms as the Bach Chorales Multiple Chord Labels (BCMCL) dataset (Section 6.5). These parallel annotations will enable us to explore multi-label label and label distribution learning for automatic chord labelling, which will be introduced in Chapter 7 as a way to address ambiguities in automatic chord labelling.

Chapter 7 Building automatic chord labellers using multi-label learning and label distribution learning

As discussed in Chapter 1, the ambiguity in chord labelling induces multiple possible analyses, and the ones prepared by different experts can often be different, as indicated in Section 4.2.1. In automatic chord labelling:

- Is it possible to train automatic chord labellers to generate multiple alternate analyses simultaneously? If so, how?
- If there are multiple possible correct labels, are some still better fit than others? Can this be determined, quantified, and automated? If so, how?

The research introduced in this chapter will answer these two research questions, which were originally proposed in Section 1.6. Based on the introduction of the supervised machine learning paradigms introduced in Chapter 2, I will use multi-label learning and label distribution learning to respectively address the first and the second research questions. These two paradigms will enable automatic chord labellers to generate multiple parallel labels for each chord, either in the form of binary labels or a distribution of probabilities, respectively. They are particularly useful in chord labelling, since the former will inform machines of the collection of possible chord labels, and the latter will further inform machines to which degree each chord label is favoured.

This chapter is organized as follows: I will first propose a modified version of the Bach Chorales Multiple Chord Labels (BCMCL) dataset in Section 7.1, which incorporates the annotations of a new rule-based Algorithm E that labels chords based on the musical surface and adds more varieties of chord interpretations to the initial version of BCMCL (Section 6.5). Then, the experiments on multi-label learning and label distribution learning will be respectively introduced in Section 7.2 and Section 7.3, following the discussions in Section 7.4, and the conclusions of the work in this chapter will be given in Section 7.5.

7.1 Data: the modified BCMCL dataset, BCMCL 1.1

7.1.1 Algorithm B′ and Algorithm E

In Section 6.5, I introduced the new Bach Chorales Multiple Chord Labels (BCMCL) dataset, which includes 120 chorales annotated with the chord labels output by each of the four rule-based algorithms (A, B, C, and D) described in Section 6.4.2. Based on Bach's figured bass, Algorithm A and B generate chord labels that are faithful to the harmony theory in 18th-century treatises, when figured bass was popular, and Algorithm C and D aim to generate chord labels based on modern harmony theory. In this chapter, the BCMCL dataset will be used to explore multi-label learning and label distribution learning, so that the resulting models can label chords based on the musical surface automatically without the figured bass.

Let us look at some chord label annotations from BCMCL in Figure 7-1. As introduced in Section 6.4.2.1, Algorithm A labels chords solely based on figured bass (without considering the upper voices), and there will be no label for slices without figures. Since many Baroque composers tend to omit figures for root position major or minor triads, these triads will not be labelled by Algorithm A (e.g., m. 3.1 of Figure 7-1). Therefore, Algorithm B was proposed in Section 6.4.2.2 to label such chords with the consideration of the musical surface, but it (along with Algorithm A) will sometimes label a slice with "?", meaning the chord of the slice is undetermined and does not have one of the nine candidate qualities considered: Major, minor, diminished, and augmented triads, as well as major, minor, dominant, half-diminished, and fully diminished 7^{th} chords.

Since I aim to find complete chord label annotations for this study, the annotations from Algorithm A and Algorithm B are not suitable since they both include "?", as some chords are undetermined. Therefore, the annotations from Algorithm A and Algorithm B will not be used for this study. Instead, I will create a variant of Algorithm B, Algorithm B′, which will replace "?" with the chord label from the subsequent slice, just as is done by Algorithm C introduced in Section 6.4.2.3, where it could be a suspension being resolved (m. 4.2 of Figure 7-1), or with a retarded bass (m. 8.2 of Figure 7-1). I will incorporate the chord label annotations of Algorithm B′, Algorithm C, and Algorithm D for this study.

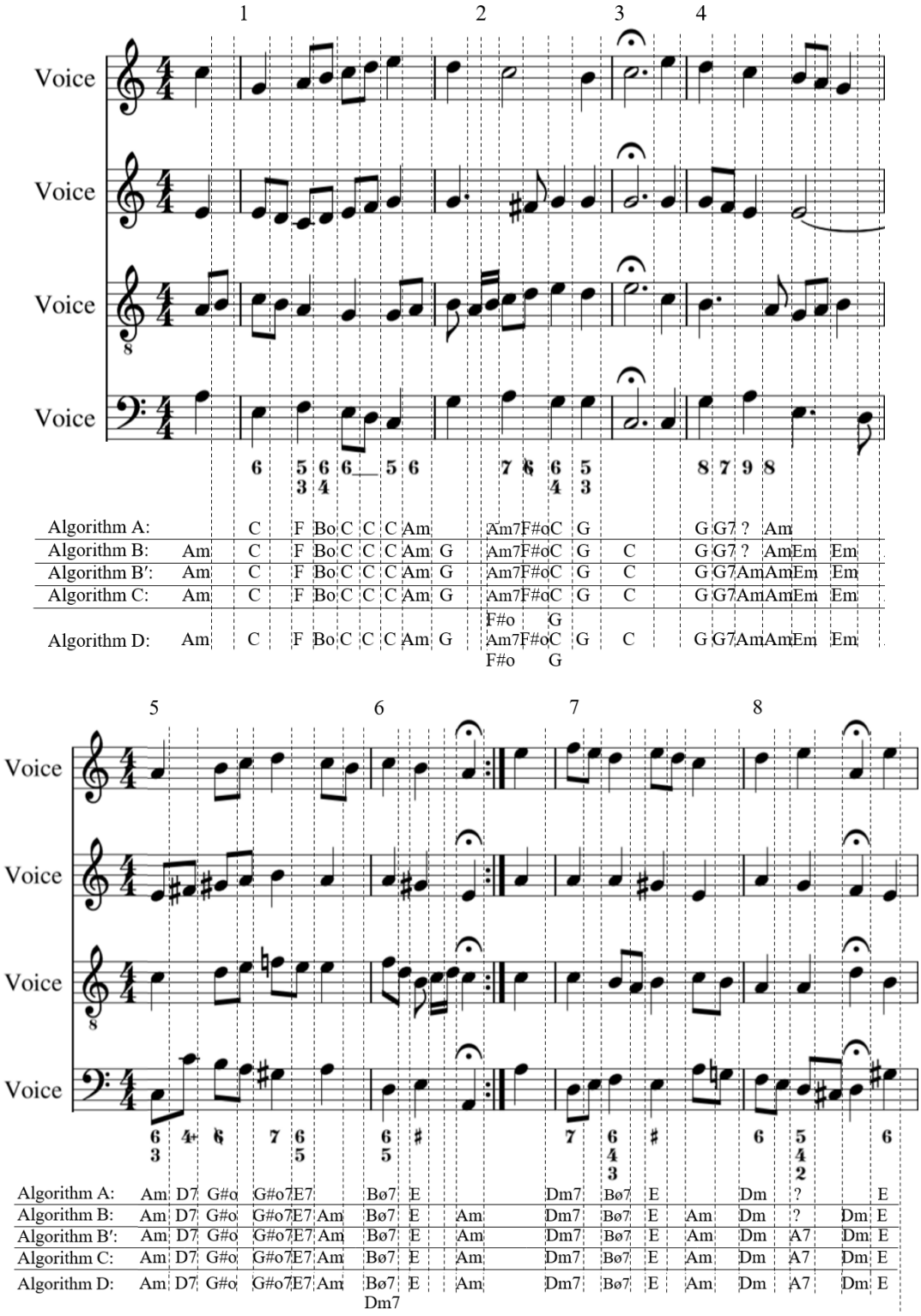


Figure 7-1: The first measures of BWV 33.06 "Allein zu dir, Herr Jesu Christ" from the Bach Chorale Multiple Chord Labels (BCMCL) dataset. The parallel tracks of chord labels from the four algorithms (Algorithms A, B, C, and D) are attached at bottom. Algorithm B' is also added as a comparison to Algorithm B.

The aim of this chapter is to generate multiple labels for chords. However, it turns out that the algorithm with the most labels per chord, Algorithm D, has less than 5% of its slices with more than one annotation (see Table 6-2). In order to create more annotations per slice, as it may happen when multiple annotators were labelling chords (see the statistics at Section 4.2.1), I propose Algorithm E, where chord labels can be generated solely based on the musical surface, whereas the chord labels generated by Algorithm B′, Algorithm C, and Algorithm D are all based on the combination of figured bass and the musical surface. The example of Algorithm E's output is illustrated in Figure 7-2, and the analytical strategy of Algorithm E can be described as follows:

- First, the algorithm will examine the sonority (pitch-class set) of each slice in the musical surface sequentially. If the pitch class set of the sonority conforms to any of the nine candidate chord qualities (major, minor, diminished, and augmented triads, as well as major, minor, dominant, half-diminished, and fully diminished 7th chords), the corresponding chord label will be associated with the slice, and no label will be output otherwise. For example, the sonority of m. 1.1 of Figure 7-2 is {C, E, G}, therefore it is a C chord; the sonority of m. 1.1.5 {E, G, B, D} will be an Em7 chord; the sonority of m. 4.2 {A, B, E, C} will not result in any chord since it does not conform to any of the nine candidate chord qualities.[94]
- The algorithm then incorporates the two ways of labelling suspensions with chords, as was done by Algorithm C (introduced in Section 6.4.2.3): For the "7–6", "6–5", and cadential "6/4" suspensions[95] (established by either "4–3" or "6/4–5/3" suspensions), the suspended notes can be either treated as chord tones, resulting in, respectively, a 7th chord, a 6th chord, or a 6/4 chord, or they can be treated as non-chord tones, in which case they should adopt the chord label from the slice to which the suspension is resolved. Therefore, slices with any of these three kinds of suspensions (e.g., m. 2.2 and m. 2.3 of Figure 7-2) will have two possible chord labels.

Overall, the analytical strategy of Algorithm E is rather basic: Non-chord tone interpretation is generally not considered, except for suspensions and the sonorities which do not conform to any of the nine chord qualities considered. This strategy is not presented as a high-

[94] This process of mapping sonority (pitch class set) into chord label is performed by the `music21.chord.Chord` object in Python.

[95] Here, an en dash (–) represents the voice leading motion and a slash (/) separates multiple figures within a single figure.

quality chord labelling algorithm. It serves more to generate synthetic data that adds more varieties to chord label interpretations to facilitate this research. Overall, Algorithm E can output zero to two chord labels for each slice. The annotations of Algorithm B′, Algorithm C, Algorithm D, and Algorithm E are shown in Figure 7-2, and they will be referred to as *the modified BCMCL dataset*, or *BCMCL 1.1*, the original BCMCL dataset will be referred to as *BCMCL 1.0*, from now on.

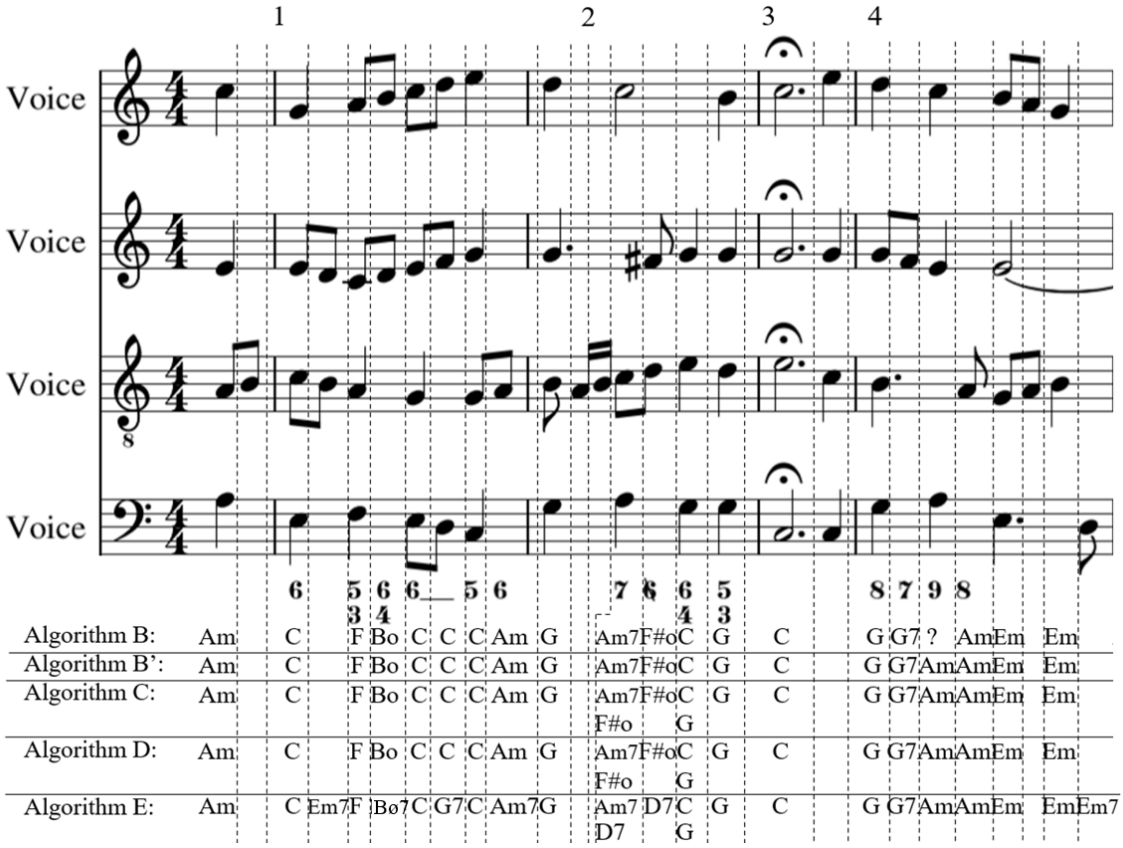
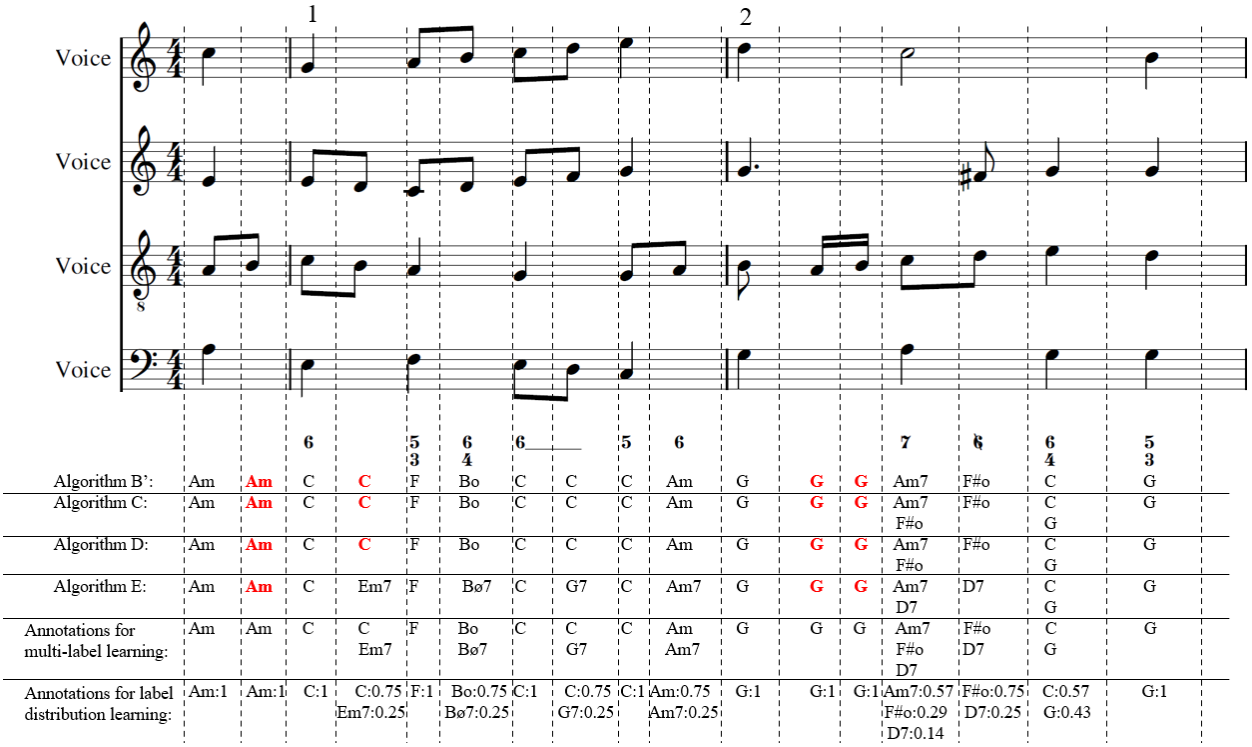


Figure 7-2: The first four measures of BWV 33.06 "Allein zu dir, Herr Jesu Christ" from the Bach Chorale Figured Bass (BCFB) dataset. Figured bass annotations are shown below the bass line along with the chord labels produced by Algorithm B, Algorithm B', Algorithm C, Algorithm D, and Algorithm E, separated by horizontal lines. Algorithm B is shown as a comparison to Algorithm B' and is not included in BCMCL 1.1.

7.1.2 Preprocessing of BCMCL 1.1

Before using BCMCL 1.1 for multi-label learning and label distribution learning, the chord label annotations of BCMCL 1.1 need to be pre-processed to comply with the required data formats for both paradigms. As shown in Figure 7-3, the annotations for multi-label learning are the union

of all unique chord labels from Algorithm B′, C, D, and E for each slice. For label distribution learning, it further counts the number of votes each chord label gets from all four algorithms, then the numbers are normalized as probabilities to serve as the membership scores for each of all the chords. Take the slice of m. 2.2 as an example, it contains three unique chord labels: Am7, F♯o, and D7 as the annotation for multi-label learning. The numbers of votes these three chord labels get from all four algorithms are: Four votes for Am7, two votes for F♯o, and one vote for D7.[96] Then, these numbers are divided by seven, the sum of all the votes from the four algorithms, which results in probabilities as membership scores for each of the three chord labels.



			6		5/3	6/4	6		5	6				7	6	6/4	5/3
Algorithm B':	Am	Am	C	C	F	Bo	C	C	C	Am	G	G	G	Am7	F#o	C	G
Algorithm C:	Am	Am	C	C	F	Bo	C	C	C	Am	G	G	G	Am7 F#o	F#o	C G	G
Algorithm D:	Am	Am	C	C	F	Bo	C	C	C	Am	G	G	G	Am7 F#o	F#o	C G	G
Algorithm E:	Am	Am	C	Em7	F	Bø7	C	G7	C	Am7	G	G	G	Am7 D7	D7	C G	G
Annotations for multi-label learning:	Am	Am	C	C Em7	F	Bo Bø7	C	C G7	C	Am Am7	G	G	G	Am7 F#o D7	F#o D7	C G	G
Annotations for label distribution learning:	Am:1	Am:1	C:1	C:0.75 Em7:0.25	F:1	Bo:0.75 Bø7:0.25	C:1	C:0.75 G7:0.25	C:1	Am:0.75 Am7:0.25	G:1	G:1	G:1	Am7:0.57 F#o:0.29 D7:0.14	F#o:0.75 D7:0.25	C:0.57 G:0.43	G:1

Figure 7-3: The first two measures of BWV 33.06 "Allein zu dir, Herr Jesu Christ" from the Bach Chorale Figured Bass (BCFB) dataset, with the chord labels produced by Algorithm B', Algorithm C, Algorithm D, and Algorithm E, shown below the bass line and separated by horizontal lines. The chord labels in red mean that they are not originally produced by the algorithms but inherited from the previous slice that has a definitive chord label. The resulting annotation for multi-label learning is the union of all possible chord labels for each slice. For label distribution learning, it further counts the number of votes each chord label gets from all four algorithms, then the numbers are normalized as probabilities to serve as the membership scores for each of all the chords shown above.

[96] The weight of each vote from all four algorithms are treated equally as of now. It will be interesting to experiment assigning different weights to these algorithms in the future.

7.1.3 Statistics of BCMCL 1.1

In Section 7.1.1, I proposed Algorithm E that generates chord labels based solely on the musical surface, a different analytical strategy than Algorithm B′, Algorithm C, and Algorithm D. Here, I will show the statistics of BCMCL 1.1 and compare them to those of the original BCMCL (BCMCL 1.0) from Section 6.5.

Table 7-1: The number of chorales, note onset slices, candidate chord qualities, chord types (identified by the combination of chord root and quality, see Glossary), and unique chord labels (including all labels for all slices produced by Algorithm D and Algorithm E) in the BCMCL 1.1 dataset and the original BCMCL dataset (BCMCL 1.0) from Section 6.5. Total slice counts and percentages (divided by the number of note onset slices) are also provided for slices with two, three, and four possible chord labels.

Category	**Number from BCMCL 1.1**	**Number from BCMCL 1.0**
Chorales	120	120
Note onset slices	9,617	9,617
Candidate chord qualities	9	9
Chord types	115	109
Unique chord labels	12,618 (from Algorithm D and E)	10,092 (from Algorithm D)
Slices with two chord interpretations	2,714 (28.22%)	471 (4.90%)
Slices with three chord interpretations	139 (1.45%)	3 (0.03%)
Slices with four chord interpretations	3 (0.03%)	0

Table 7-1 summarizes certain statistics on BCMCL 1.1 based on the unique chord labels produced by Algorithm D and Algorithm E. As discussed in Section 6.4.2.4, Algorithm D is the superset of Algorithm B (also B′) and Algorithm C, therefore combining it with Algorithm E will demonstrate the variety of multiple chord interpretations BCMCL 1.1 has. These statistics are also compared to those of BCMCL 1.0 from Table 6-2 side by side. The comparison shows that the percentage of slices with two-chord interpretations increases significantly from 4.90% in BCMCL

1.0 to 28.22% in BCMCL 1.1, and there are also more slices with three-chord interpretations in BCMCL 1.1 (1.45%) compared to BCMCL 1.0 (0.03%), and even four-chord interpretations found in BCMCL 1.1. The distributions of chord types and chord qualities of BCMCL 1.1 are respectively shown in Figure 7-4 and Figure 7-5, in comparison to those of BCMCL 1.0.

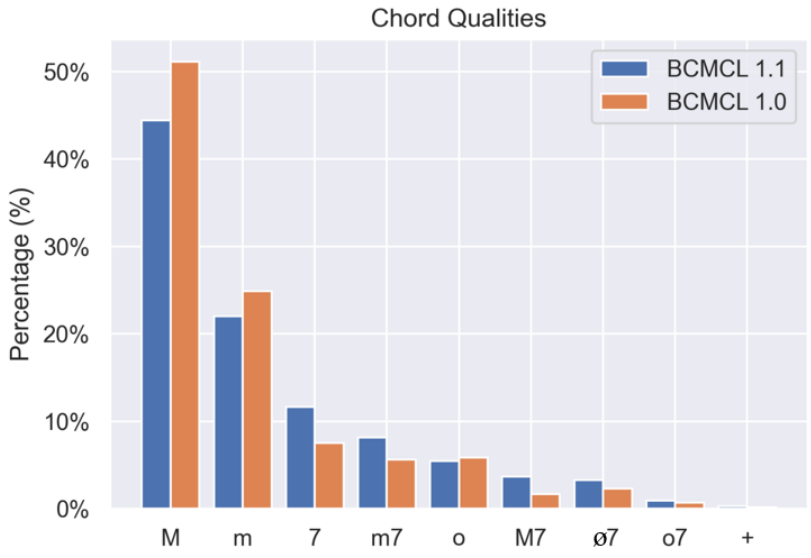


Figure 7-4: Distributions of chord qualities for BCMCL 1.1, in comparison to those of BCMCL 1.0.

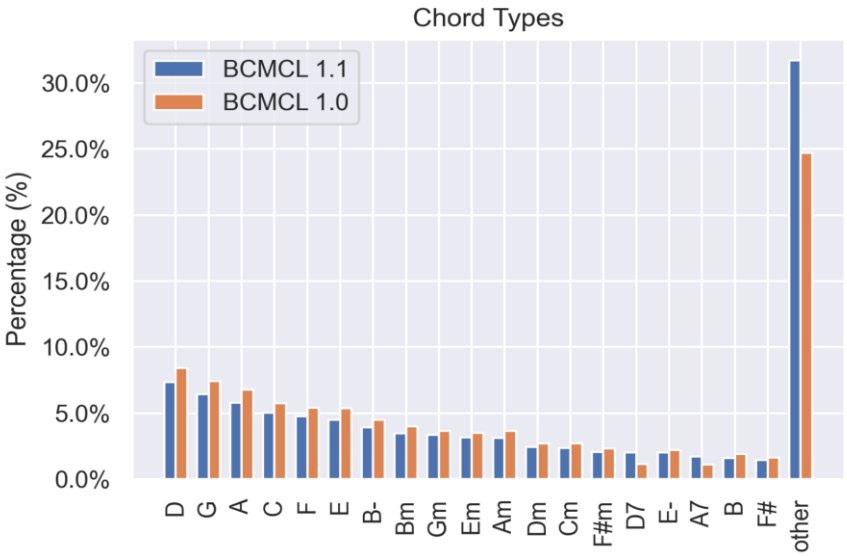


Figure 7-5: Distributions of chord types for BCMCL 1.1, in comparison to those of BCMCL 1.0.

7.1.4 Summary

In this section, I introduced BCMCL 1.1, which is an iteration of the original BCMCL 1.0 dataset and contains the definitive chord label annotations from Algorithm B′, Algorithm C, Algorithm D, and Algorithm E. Algorithm E aims to label chords based on the musical surface only and presents a different analytical strategy than Algorithm B′, Algorithm C, and Algorithm D which all label chords based on the combination of figured bass and the musical surface. As shown in Section 7.1.3, the chord labels from Algorithm E adds more possible, alternate analyses, resulting in significantly more slices with multiple chord interpretations compared to the ones of BCMCL introduced in Section 6.5.1. The annotations of BCMCL 1.1 are now released and made publicly available at https://github.com/juyaolongpaul/BCMCL/releases/tag/1.1. These annotations can serve various research purposes:

- They can serve as an interesting source for the empirical study of chord labelling ambiguity. For example, what characteristics of the musical surface and figured bass will lead to ambiguity? Specifically, is the number of possible chord labels and the type of chord labels related to some of these characteristics? What is the relationship among all the parallel chord labels? Are they in fact similar or different?
- They can also serve as a preliminary example of how chord labelling annotations can vary among different people, since the algorithms (B′, C, D, and E) that generate these annotations can be understood as a group of four different analysts, each with a different analytical strategy. Therefore, the annotations in Figure 7-3 of multi-label learning represent the collection of possible chord labels from these "analysts", and the ones of label distribution learning can be understood as how much these labels are voted and favoured by the group of these analysts. Exploring multi-label learning and distribution learning on BCMCL 1.1 can be intriguing since it aims to teach machines to learn from not only one but multiple analysts, particularly on all the possible analyses and how much each one is favoured. These studies will be presented in Section 7.2 and Section 7.3, respectively.

7.2 Multi-label learning

In this section, I will use BCMCL 1.1 to explore multi-label learning for automatic chord labelling, so that the model can generate multiple alternate labels for each chord based on a variety of different analytical perspectives.

7.2.1 Data

The data for multi-label learning is obtained using the preprocessing of BCMCL 1.1 introduced in Section 7.1.2 and shown in Figure 7-3. There are 120 chorales, with a total number of 9,617 slices. These chord label annotations are represented either as a one-hot vector, if there is only one chord label; or a multi-hot vector, if there are two or more chord labels. The input features are also represented as a multi-hot vector. I used the feature combination from Section 5.2.2.2 as a basis, which contains four different kinds of features: **PC12**, **M**, **O**, and **Wn**. To summarize:

1. **PC12**: A 12-D binary vector of enharmonic pitch classes present in the slice.
2. **M**: A 3-D indication of the metrical context of the slice, which specifies whether a slice occurs on the downbeat of a measure, on another whole beat (e.g., beat 2, 3, or 4 in 4/4), or on a fractional beat (e.g., beat 3.5).
3. **O**: A 12-D vector indicating which PC12 pitch classes are real onsets and which are artificial onsets.
4. **Wn**: A variable size vector containing the (non-Wn) features from the n previous and n following slices (e.g., W1 indicates that features for the directly preceding and directly following slices are included in the features of the current slice). These surrounding slices are called “contextual windows”.

Here, I use **W2** as contextual windows, which indicates that features for the two directly preceding and two directly following slices are included in the features of the current slice. I also added the following additional feature: **B**, which is a 12-D binary vector of enharmonic pitch classes present in the bass voice, since the bass note may be helpful for chord labelling. See Figure 7-6 for the specific encodings of these features with a music example.

I also implement data augmentation (abbreviated as **A**, adopted by the PC12MOW2A and PC12MOW2BA configurations specified in Table 7-2) by transposing each chorale and the corresponding chord labels to all 11 other possible enharmonic keys, resulting in an augmented dataset 12 times (a total of 115,404 slices) as big as the original one, and there are 177 chord types in the augmented chord labels. When data augmentation was not used, such as PC12MOW2 and PC12MOW2B in Table 7-2, all chorales and corresponding chord labels were transposed to the key of either C major or A minor, depending on the original mode, to make the tonal relationships between pitch classes consistent across the dataset.[97] There are 62 chord types in the transposed chord labels.

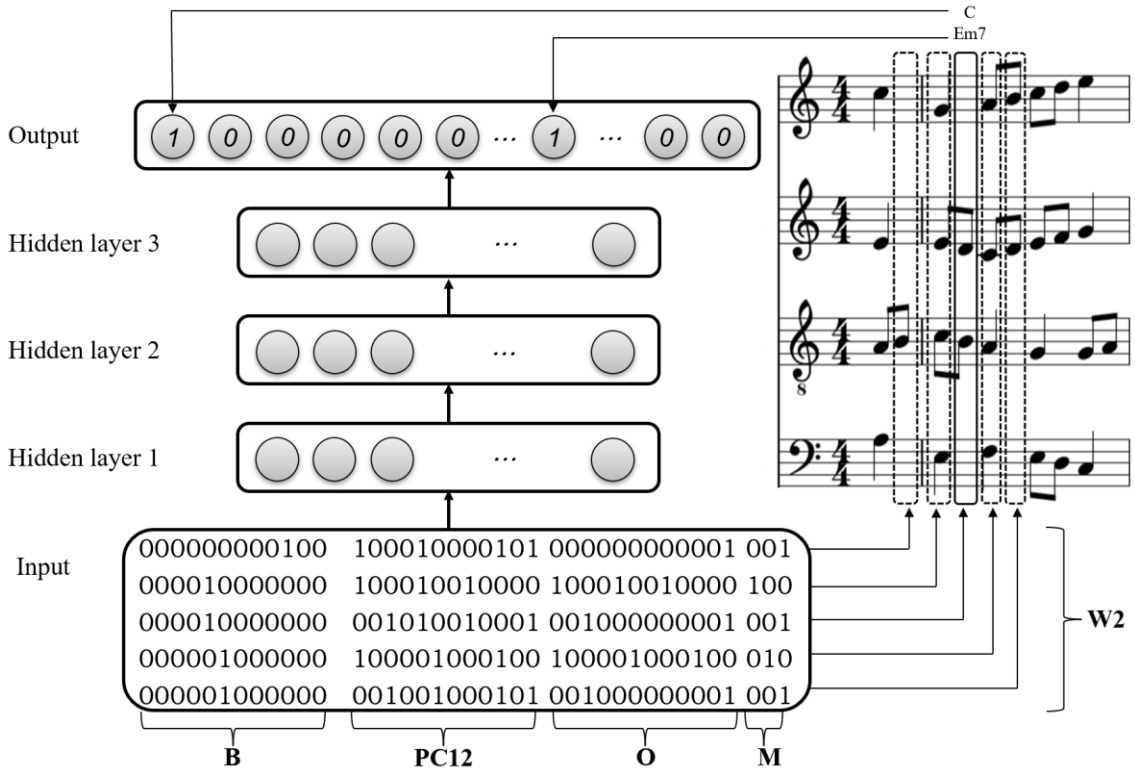


Figure 7-6: Illustration of the PC12MOW2B multi-label learning model introduced in Section 7.2.2, using the first measure of BWV 33.06 "Allein zu dir, Herr Jesu Christ" as an example shown on the right. The slice with the solid line rectangle is the current slice, whose input features are connected with an arrow. Its two chord-label annotations are also indicated in the corresponding bits of the output vector with arrows. The features of the preceding and the following slices (with dashed line rectangles) are concatenated as context in the input vector.

[97] The built-in key transposition function from music21 was used, with the Aarden-Essen key profile (https://web.mit.edu/music21/doc/moduleReference/moduleAnalysisDiscrete.html#aardenessen).

7.2.2 Experimental setup

As discussed in Section 2.3, there are different techniques to approach multi-label learning. Here, I use the same method in Section 6.3.3.2 which involves a combination of independent binary classifiers, known as the *first-order strategy* introduced in Section 2.3.3.1. Here, each classifier is used to predict a particular chord label type, which means PC12MOW2A and PC12MOW2BA use an independent binary classifier for each of the 177 augmented chord labels, and PC12MOW2 and PC12MOW2B use an independent binary classifier for each of the 62 transposed chord labels introduced in Section 7.2.1. Compared to other alternatives from Section 2.3.3.2 and Section 2.3.3.3, this approach is efficient in handling multi-label data and is easy to implement. I use Deep Neural Networks (DNN)[98] as classifiers, and the model architecture is shown in Figure 7-6.

Ten-fold cross-validation was used for evaluation. For the DNN experiments, I divided the data into training (80%), validation (10%), and testing (10%) groups. For evaluation metrics, I used subset accuracy, micro-precision, micro-recall, and micro-f1 introduced in Section 2.3.5.2.[99] Additionally, I proposed a new evaluation metric: *Inclusive accuracy*, where the prediction is considered correct if the label set (must contain at least one chord) is identical to or a subset of the ground truth label set. For example, if the predicted chord label is ["C"] and the ground truth chord labels are ["C", "Am"], subset accuracy will consider the prediction wrong since it is not identical to the ground truth, but inclusive accuracy will consider the prediction right since the prediction is not a wrong answer, it is just not as comprehensive as the ground truth chord labels.

7.2.2.1 Significance testing setup

Of particular interest, I would like to see whether the use of the bass voice feature **B** (comparing PC12MOW2 against PC12MOW2B and PC12MOW2A against PC12MOW2BA) and data augmentation **A** (comparing PC12MOW2 against PC12MOW2A and PC12MOW2B against PC12MOW2BA) results in a significant performance boost. Before discussing the results, I would

[98] I used a feedforward network with three hidden layers, each with 300 hidden units. Adaptive Moment Estimation was used as an optimizer, with binary cross-entropy as loss function.
[99] Micro-precision, micro-recall, and micro-f1 are noted as $MicroPrecision, MicroRecall, and\ MicroF1$ in Section 2.3.5.2.

like to present the significance test that will be used to determine the effects of **B** and **A**. I will use an *independent two-sample t-test*,[100] which compares one variable (i.e., **B** or **A** in this study) between two groups. This test makes three assumptions that are not necessarily met in the analysis performed here: (1) The two groups being compared are independent, (2) the samples in each group are normally distributed, and (3) the variances from two groups are considered homogeneous. The first assumption (independence) is not met, given that cross-validation is involved, and the remaining two assumptions have not been tested. I nonetheless used the *independent two-sample t-test* here as a preliminary approximation of statistically valid significance testing. This test is also conducted in Section 7.3.3.

Here, I will use **B** (comparing the performances of PC12MOW2 against PC12MOW2B) as an example to explain the process of this t-test below. First, a hypothesis is needed, and I will use the *null hypothesis*, claiming that the use of **B** (PC12MOW2B) does not result in a significant performance boost over the absence of **B** (PC12MOW2). To verify this hypothesis, two values need to be calculated and defined: (1) The p-value, which reflects the *probability* of the null hypothesis being true and will be calculated based on the provided data; (2) the significance level α, which is the threshold suggesting that the null hypothesis will be rejected if $p < \alpha$, or accepted if $p \geq \alpha$. Much of the explanation for this significance test can be found in Pernet (2015).

Here, I use the mean value, standard error of the mean, and a sample size of 10 (the number of cross-validation folds) as input, and I choose $\alpha = 0.05$ as the significance level, meaning if the calculated $p < 0.05$, the null hypothesis will be rejected and the use of **B** corresponds to a statistically significant performance boost over the absence of **B**. There are five evaluation metrics used in Table 7-2, therefore verifying the use of **B** involves five experiments: Comparing the performances of PC12MOW2 against PC12MOW2B in each of the five metrics (subset accuracy, micro-precision, micro-recall, and micro-f1, and inclusive accuracy). The same process is also applied to the comparison between PC12MOW2A and PC12MOW2BA, resulting in another five experiments. Therefore, there are altogether 10 experiments, and if $p < 0.05$ is achieved in all 10 tests, I will conclude that the use of **B** is *overall effective*, resulting in a consistent performance boost at all regards; if $p < 0.05$ is achieved only under some (at least one but less than five) metrics, I will conclude that the use of **B** is *partially effective* and mention the corresponding

[100] The tool available at: https://www.graphpad.com/quickcalcs/ttest1.cfm.

metric(s); if $p < 0.05$ is not achieved under any metric, I will conclude that the use of **B** is *not effective*. This process is also applied to the use of **A**. Additionally, I am also interested to see whether the use of the combination of **B** and **A** results in a performance boost. This will involve the comparison of the performances of PC12MOW2 and PC12MOW2BA under the five metrics, and I will conclude whether the use of this combination is *overall effective*, *partially effective*, or *not effective* using the method shown above.

As a general note, if any of the following discussions uses the phrase *significantly higher* or *significantly lower*, it means that the corresponding two groups of results have undergone an *independent two-sample t-test*, yielding a value of $p < 0.05$, albeit in circumstances where the assumptions of test are not fully met, as noted above.

7.2.3 Results

The results are shown in Table 7-2, which shows the performances of the multi-label learning models on BCMCL 1.1. An example of multi-label prediction and its comparison to the ground truth is shown in Figure 7-7. When combining the results from Table 7-2 and Figure 7-7, it seems that the model might under-predict, compared to the labels from the ground truth. This observation is further supported by comparing the performances of subset accuracy against those of inclusive accuracy in Table 7-2, where I observe a significant increase in numbers. This means that there is a considerable number of test examples whose predicted chord labels are not identical to the ground truth but nonetheless are the subset of it. This observation is also supported by combining the performances of micro-precision and micro-recall, where the former is significantly higher than the latter, suggesting that when the model predicts, it is relatively accurate, with a score of 0.920 micro-precision using the PC12MOW2A model, but the prediction does not cover as many chord labels in the ground truth, with only a score of 0.816 from the same model.

After conducting all the significant tests introduced in Section 7.2.2.1, it shows that **B** is *not effective,* and **A** is *partially effective* under the *inclusive accuracy* and *micro-precision* metrics; the combination of **B** and **A** is *partially effective* under the *subset accuracy* and *micro-F1* metrics.

Table 7-2: Multi-label learning models' performances on BCMCL 1.1. Columns indicate evaluation metrics for multi-label learning and rows indicate model configurations (see Section 7.2.1). Uncertainty values show standard error across cross-validation folds.

Model Configuration	Subset Accuracy	Inclusive Accuracy	Micro-precision	Micro-recall	Micro-F1
PC12MOW2	73.8±0.7%	89.0±0.5%	0.901±0.005	0.810±0.005	0.853±0.004
PC12MOW2A	75.6±0.8%	91.0±0.7%	0.920±0.006	0.816±0.007	0.865±0.006
PC12MOW2B	74.5±0.7%	88.1±0.4%	0.897±0.003	0.822±0.005	0.858±0.004
PC12MOW2BA	76.3±0.8%	90.2±0.6%	0.914±0.006	0.828±0.007	0.869±0.006

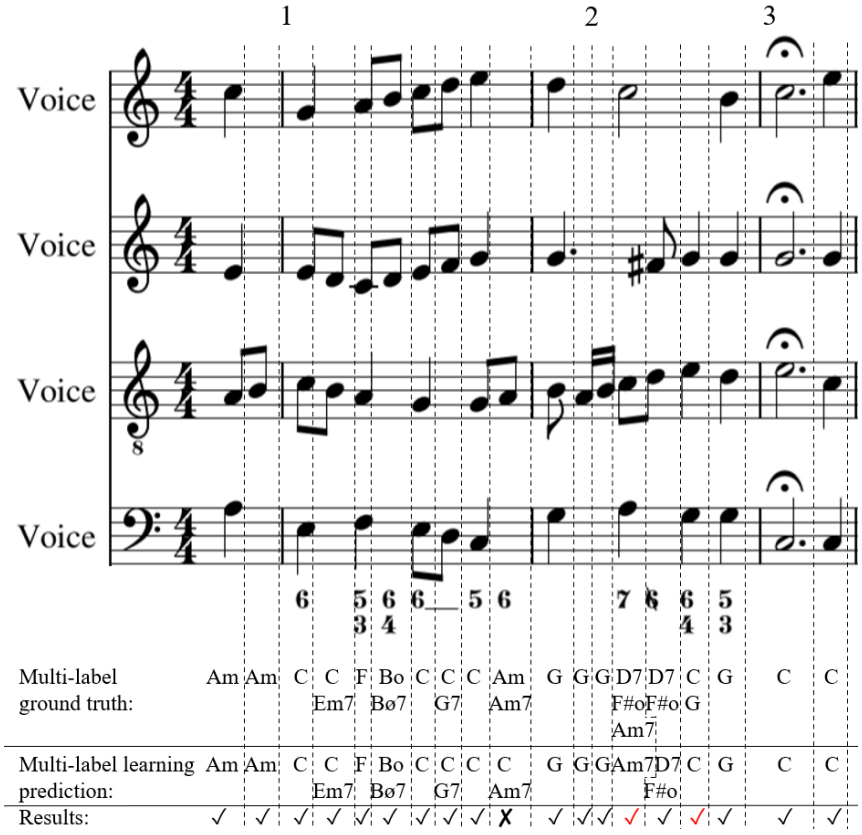


Figure 7-7: The first three measures of BWV 33.06 "Allein zu dir, Herr Jesu Christ" from the Bach Chorale Figured Bass (BCFB) dataset, where the ground truth chord labels and the predicted chord labels generated by PC12MOW2BA are shown below the bass line. If they are identical, the result will be "✓", meaning the prediction is considered correct by both subset accuracy and inclusive accuracy; if the prediction is the subset of the ground truth (e.g., m. 2.2 and m. 2.3), the result will be "✓", meaning the prediction is considered correct by inclusive accuracy only, if the result is "✗". it means the prediction is considered wrong by both subset accuracy and inclusive accuracy.

7.3 Label distribution learning

In this section, I will use BCMCL 1.1 to explore label distribution learning for automatic chord labelling. As discussed in Section 2.4, it is a more general case of multi-label learning, where each slice is not only associated with multiple chord labels, but also the membership scores show how much each chord label describes the example, relatively speaking. The resulting model can automatically generate label distributions. The data used for label distribution learning will be introduced in Section 7.3.1, the experimental setup and results will be introduced in Section 7.3.2 and Section 7.3.3, respectively.

7.3.1 Data

The data that will be used to train and evaluate label distribution learning models is similar to that of multi-label learning in Section 7.2.2, except that the output changes from a multi-hot binary vector to a real-value vector, where each dimension indicates the membership score for each chord type using the preprocessing of BCMCL 1.1 introduced in Section 7.1.2. Take the slice m. 2.2 of Figure 7-3 as an example, the output vector of this particular case will have three positive values: 0.57, 0.29, and 0.14 for the bits of the chord labels Am7, F♯o, and D7, respectively, and the rest of the vector is set to zero.

7.3.2 Experimental setup

As discussed in Section 2.4.1, there are also different techniques to approach label distribution learning. Here, I also use Deep Neural Networks (DNN)[101] as classifiers, and the model architecture is shown in Figure 7-8. This architecture is similar to that of multi-label learning shown in Figure 7-6, but there are two major differences:

1. The output vector now stores real values between [0, 1] (all sum up to 1) as the membership score for each chord label, while in multi-label learning, these values are binary (either 0 or 1).

[101] I used a feedforward network with three hidden layers, each with 300 hidden units. Adaptive Moment Estimation was used as an optimizer.

2. The loss function now is the Kullback-Leibler divergence (see Section 2.4.2) to reduce the difference between the predicted and the ground truth label distributions, while in multi-label learning, it is binary cross-entropy.

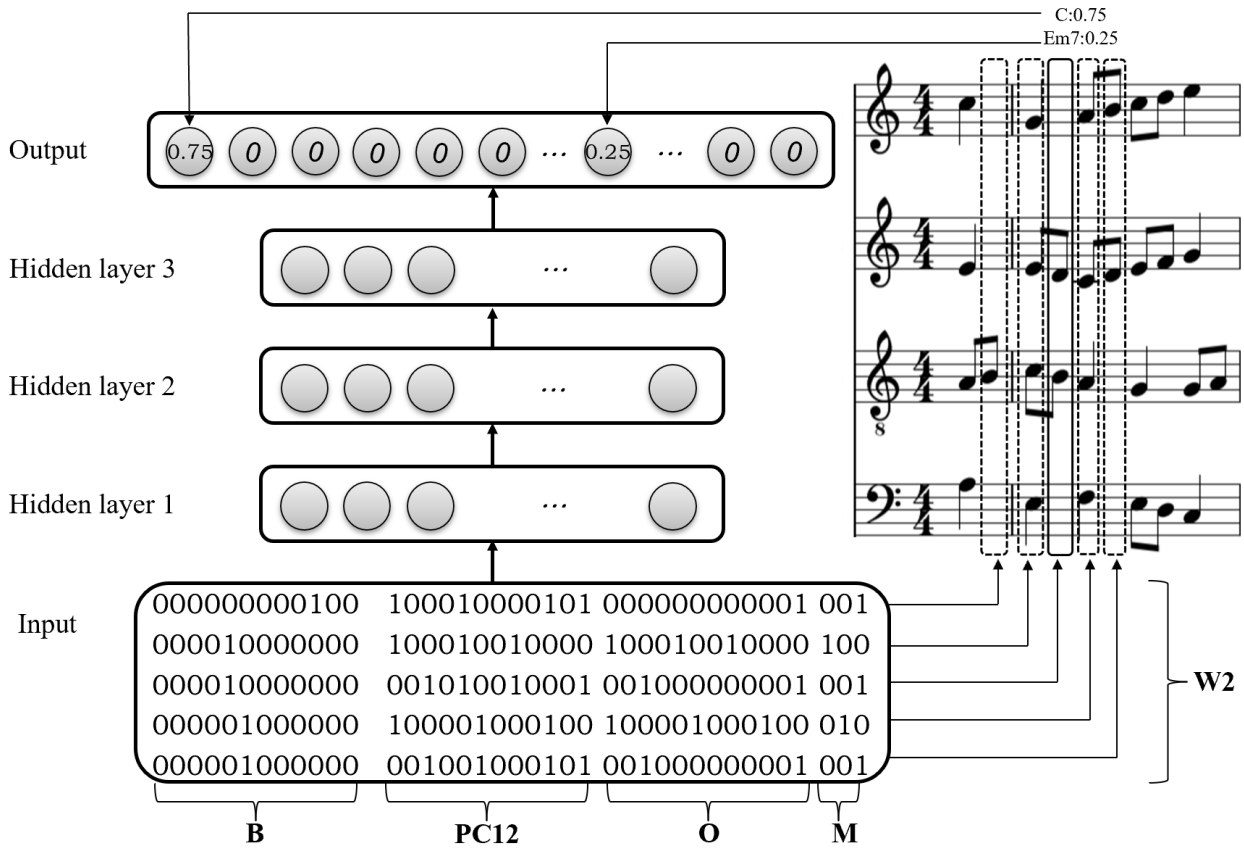


Figure 7-8: Illustration of the PC12MOW2B label distribution learning model introduced in Section 7.3.2, using the first measure of BWV 33.06 "Allein zu dir, Herr Jesu Christ" as an example shown on the right. The slice with the solid line rectangle is the current slice, whose input features are connected with an arrow. Its two chord-label annotations with membership scores are also indicated in the corresponding bits of the output vector with arrows. The features of the preceding and the following slices (with dashed line rectangles) are concatenated as context in the input vector.

Here, I also used the same four kinds of model configurations (PC12MOW2, PC12MOW2A, PC12MOW2B, and PC12MOW2BA) discussed in Section 7.2.3. Ten-fold cross-validation was used for evaluation, and I once again divided the data into training (80%), validation (10%), and testing (10%) groups. For evaluation metrics, I also use the Kullback-Leibler divergence, which has been used in the recent literature of label distribution learning (Geng 2016; Gao

et al. 2017). A lower value of Kullback-Leibler divergence indicates a better performance and shows a closer resemblance between the predicted label distribution and the ground truth. Furthermore, I also propose a new evaluation metric: *Ranking accuracy*, where the prediction is considered correct if the order of chord labels ranked by its value (in a descending order) is identical to that of the ground truth, as shown in Figure 7-9. It is a useful evaluation metric besides the Kullback-Leibler divergence, since it reflects the percentage of slices whose predicted chord labels are in the same order as the ground truth, regardless of how the actual values may differ.

7.3.3 Results

Table 7-3: Label distribution learning models' performances on BCMCL 1.1. Columns indicate evaluation metrics and rows indicate model configurations (see Section 7.2.1). Uncertainty values show standard error across cross-validation folds. For Kullback-Leibler divergence, a lower value indicates a better performance and shows a closer resemblance between the predicted label distribution and the ground truth, and vice versa.

Model Configuration	Kullback-Leibler Divergence	Ranking Accuracy
PC12MOW2	0.647±0.030	74.1±0.6%
PC12MOW2A	0.592±0.023	75.7±0.7%
PC12MOW2B	0.621±0.029	75.0±0.7%
PC12MOW2BA	0.548±0.021	76.7±0.7%

The results of label distribution learning models are shown in Table 7-3. The *Independent two-sample t-test* introduced in Section 7.2.2.1 is also used here. Results show that **B** and **A** are both *not effective*, but the combination of **B** and **A** is *overall effective* (under both Kullback-Leibler divergence and ranking accuracy metrics).

An output example of the label distribution learning model is shown in Figure 7-9, which indicates that the model is able to automatically predict and quantify the membership score, or the fitness of each chord type.

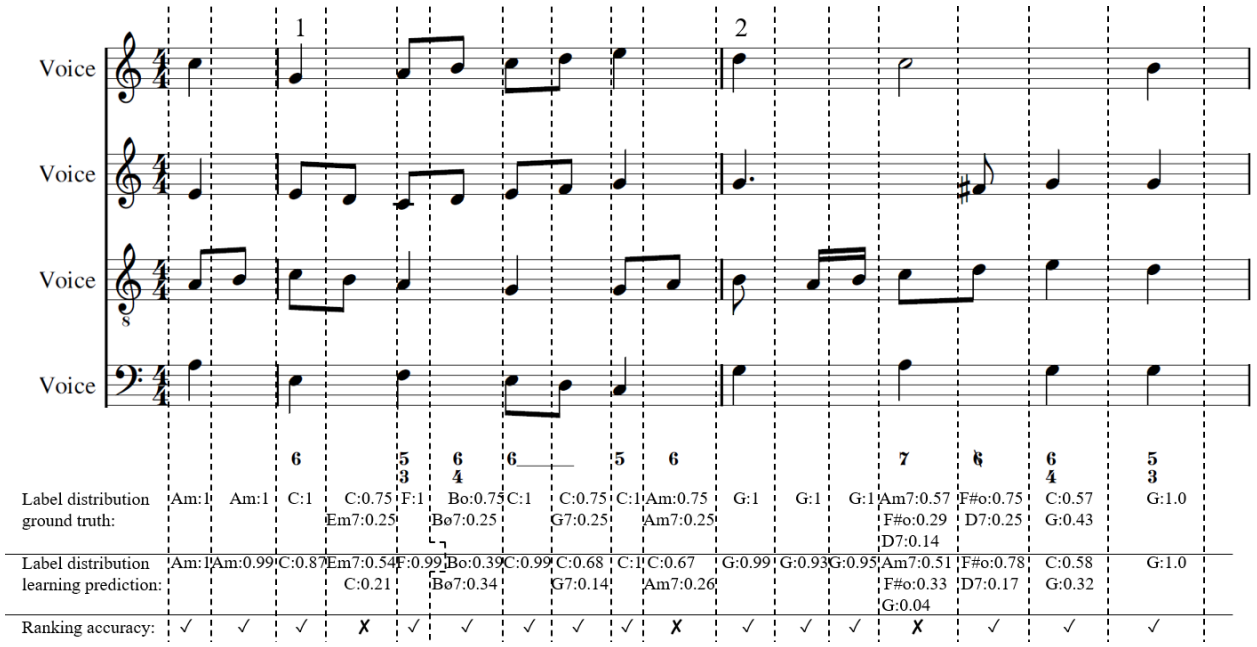


Figure 7-9: The first two measures of BWV 33.06 "Allein zu dir, Herr Jesu Christ" from the Bach Chorale Figured Bass (BCFB) dataset, where the label distribution ground truth, the PC12MOW2BA label distribution learning model's prediction, and the results of ranking accuracy are shown below the bass line.

7.4 Discussion

The work introduced in this chapter so far has been an exploration of teaching machines how to deal with ambiguity in automatic chord labelling, when multiple possible analyses are available. I used multi-label learning and label distribution learning as an attempt to enable machines to learn from these parallel analyses and see how much they are able to predict the analyses similar to those from BCMCL 1.1, which I created to reflect what chord label annotations can be generated by a group of four algorithms, each with a different analytical strategy. The research for these two supervised machine learning paradigms has been introduced in respectively Section 7.2 and Section 7.3. When combining all the results and observations from both sections, a few general trends can be summarized as follows:

- The PC12MOW2BA multi-label learning model in Section 7.2 achieved a subset accuracy of 76.3±0.8% and an inclusive accuracy of 90.2±0.6% on BCMCL 1.1, where the latter is significantly higher than the former. It seems that the model might under-predict, compared to the labels from the ground truth, and when it predicts, it is relatively accurate, but the prediction does not cover as many chord labels in the ground truth. The PC12MOW2BA label distribution learning model in Section 7.3 achieved a ranking accuracy of 76.7±0.7%.
- This work also explored different feature combinations, specifically regarding whether the addition of the bass voice feature **B** and data augmentation **A** is effective at improving the model's performances on multi-label learning and distribution learning. Based on the preliminary approximated significance testing introduced in Section 7.2.2.1, the results show that B is overall *not effective*, and A is only *partially effective* in multi-label learning and *not effective* in label distribution learning. However, the combination of B and A is effective in both tasks, more specifically, it is *partially effective* in multi-label learning and *overall effective* in label distribution learning. As discussed in Section 7.2.2.1, these conclusions must be interpreted as preliminary indications rather than as confirmed truth.

7.5 Conclusion

The research introduced in this chapter presented another way of addressing ambiguity in automatic chord labelling by exploring two more supervised machine learning paradigms: Multi-label learning and label distribution learning,

As implementations, I first proposed a modified version of BCMCL, BCMCL 1.1 in Section 7.1, which incorporated the annotations of a new rule-based Algorithm E that resulted in 28.22% slices with multiple chord interpretations, significantly more than those of the original BCMCL introduced in Section 6.5. The experiments of multi-label learning and label distribution learning were respectively presented in Section 7.2 and Section 7.3, where deep neural networks were used as classifiers and different input feature combinations along with data augmentation were experimented. Additionally, I also proposed two new evaluation metrics in both sections: *Inclusive accuracy* and *ranking accuracy*, to respectively present the performances of the multi-label learning model and label distribution learning model in different ways which were not

captured by the existing evaluation metrics. The detailed discussions on the results of multi-label learning and label distribution learning were presented in Section 7.4.

Overall, although the chord label annotations of BCMCL 1.1 were generated using only heuristics, the automatic chord labellers achieved a subset accuracy of 76.3±0.8% and an inclusive accuracy of 90.2±0.6% on BCMCL 1.1 using multi-label learning, and a ranking accuracy of 76.7±0.7% in label distribution learning, showing promising capabilities of automatically generating multiple possible analyses, either in the form of binary labels or a distribution of membership scores for all the labels. It will be interesting to see what the performances will be when the models are applied to the data collected from human annotators in the future. As discussed in Chapter 2, there are many other techniques for multi-label learning and label distribution learning, and there are other deep learning models than DNN. Although these options have not been explored here for the time being, future research further exploring these alternatives could certainly be helpful, and the automatic chord labeller might achieve better performances in multi-label learning and label distribution learning.

Finally, to answer the two research questions introduced at the beginning of this chapter:

- Research question: *Is it possible to train automatic chord labellers to generate multiple alternate analyses simultaneously? If so, how?*
  Answer: It is possible using multi-label learning, which was explored in Section 7.2.
- Research question: *If there are multiple possible correct labels, are some still better fit than others? Can this be determined, quantified, and automated? If so, how?*
  Answer: The fitness of the multiple correct labels can be determined and quantified, for example, by counting the vote each label gets from a group of four algorithms introduced in Section 7.1.2, and dividing it using the number of all votes for normalization. It is possible to automate this process using label distribution learning, which was explored in Section 7.3.

Chapter 8 Conclusions

Overall, I believe and hope that this research introduced in this dissertation has advanced the knowledge of automatic chord labelling research, especially on how its ambiguity should be addressed. The outcome of this research can be summarized as answers to the following four research questions proposed in Section 1.6:

- Research question 1: *What is the fundamental cause of chord labelling ambiguity, and which part of the chord labelling process can result in ambiguity?*
  Answer: The fundamental cause of chord labelling ambiguity is under-specification of the process of chord labelling, which means if a clear analytical perspective is specified, there will be little ambiguity on what the chord labels should be. The following three steps of chord labelling process can result in ambiguity:
  (1) Defining chord qualities (Section 5.1.1.1);
  (2) identifying non-chord tones (Section 5.1.1.2); and
  (3) mapping chord tones into chord labels (Section 5.1.1.3).
- Research question 2: *If single-label learning is used for automatic chord labelling, what are the ways to generate single labels for chords*?
  Answer: I developed a rule-based algorithm (Section 5.1), which generates single labels for chords based on a pre-defined analytical strategy. The resulting annotations were then modified by a music theorist with more nuance, and these annotations were used to build automatic chord labellers using single-label learning (Section 5.2).
- Research question 3: *Is it possible to train automatic chord labellers to generate multiple alternate analyses simultaneously? If so, how?*
  Answer: It is possible using multi-label learning, which was explored in Section 7.2.
- Research question 4: *If there are multiple possible correct labels, are some still better fit than others? Can this be determined, quantified, and automated? If so, how?*
  Answer: The fitness of the multiple correct labels can be determined and quantified, for example, by counting the vote each label gets from a group of four algorithms introduced in Section 7.1.2, and dividing it using the number of all votes for normalization. It is possible to automatic this process using label distribution learning, which was explored in Section 7.3.

Furthermore, there are four additional insights I have for the research on automatic chord labelling, which further summarize this research and offer possible directions for future research.

8.1 Four insights

The first insight is about the role of ambiguity in chord labelling and supervised machine learning. In the existing literature of chord labelling, not much research has been found that discusses the source and characteristics of chord labelling ambiguity in depth: Why there is ambiguity in the first place, and what aspects of chord labelling can result in ambiguity. In Section 1.4, I described the process of chord labelling as three steps, where each one could result in ambiguity. This ambiguity was further discussed in Chapter 4, along with Chapter 3, where I also introduced different chord labelling theories proposed by prominent music theorists. These theories mainly contained general rules, which were underspecified and not detailed enough to provide clear directions for obtaining definitive chord label analyses.

In my opinion, this under-specification is the fundamental source of chord labelling ambiguity, leaving great room for interpretation for each individual to shape the way they conduct chord labelling. The resulting variabilities among experts' chord labelling analyses seem to be considered as individual analytical taste and identity in music theory, and comparative studies have not been performed until recently, when an increasing amount of effort has been made to automate the process of chord labelling using supervised machine learning, which relied on these annotations to build automated models (De Clercq and Temperley 2011; Ni et al. 2013; Koops et al. 2019). Some researchers believe that annotators' variabilities would present a performance upper bound for the automated models (Flexer and Grill 2016; Koops et al. 2019; Ni et al. 2013; Serra et al. 2014; Smith and Chew 2013; Flexer and Lallai 2019) in supervised machine learning, since it was hard or even unrealistic to train an automated model that agrees more than what experts agreed with each other. In summary, ambiguity in chord labelling offers a way of representing personalized analyses of each expert, but the resulting annotation variabilities among experts can be an issue in supervised machine learning, and more specifically, automatic chord labelling in the context of this dissertation.

I proposed two approaches that addressed this ambiguity issue in automatic chord labelling. One is to reduce these variabilities by specifying a clear analytical strategy, so under this strategy,

there will be only one label for each chord. The resulting annotations were then corrected by a human expert and were used to build single-label learning models for automatic chord labelling, as discussed in Chapter 5. This approach represented only one possible analysis, and the analytical strategy was well-defined, leaving no room for annotation variabilities and ambiguity under this strategy. The other approach was to incorporate these experts' variabilities as multiple parallel tracks of annotations, so that the resulting automatic chord labeller could generate not only single chord labels according to a specific analytical strategy, but also multiple labels for a single chord based on a variety of different analytical perspectives, as introduced in Chapter 7.

The second insight of this research is that the work introduced in Chapter 5, Chapter 6, and Chapter 7 required significantly less work for human analysts to provide manual annotations, which was different from most existing literature that used supervised machine learning and completely relied on manual annotations for training the automated models. The reason is that annotations provided by experts are very expensive. The research presented in this dissertation explored some possible alternatives for obtaining chord label annotations as training data.[102] In Chapter 5, a rule-based algorithm was presented to generate preliminary analyses, then a small portion of these analyses was checked and corrected by a human analyst. This approach combined the consistency of rule-based algorithms with the nuance of manual analyses to generate relatively inexpensive chord label annotations for training automatic chord labellers. In Chapter 6, I proposed four new rule-based algorithms to generate multiple parallel chord label annotations, completely forwent manual analyses/corrections, and adopted figured bass, which offered insights into the possible chord label interpretations. Therefore, I considered figured bass as a promising source of nuance, an alternative of manual annotation to obtain multiple alternate analyses, and incorporated it as an integral part of all four rule-based algorithms. Additionally, these algorithms could also be considered as automatic chord labelling models since they were entirely based on heuristics and could generate chord labels automatically, when both the musical surface and figured bass were available. In Chapter 7, I used their annotations as training data to explore multi-label learning and label distribution learning to build more automatic chord labellers.

[102] Here, I highlighted "training data" because for proper evaluations of these automatic chord labellers, a test set that contains manually annotated data by experts is still needed. Considering the training set is often considerably larger than the test set in supervised machine learning, the reduced cost of manual annotations introduced in this approach is still quite significant.

The third insight is about my speculations on how annotation variabilities may affect a model's performance. As discussed in the first insight, it seems that annotators' variabilities may present a performance upper bound for the automated models, meaning that the model's performance may not pass a certain limit, unless the variabilities or inconsistencies in the annotations are resolved. This claim seems to coincide with the findings and discussions of automatic figured bass annotation results in Section 6.3.4, where Bach's annotations exhibited a degree of variabilities under seemingly similar musical contexts. Take Figure 6-5 (b) as an example, Bach did not label the first "♯" at m. 2.3, but labelled the second one at m. 3.3, and the model predicted "♯" in both places. Both predictions were correct, but one was evaluated as wrong using Bach's annotations. In this case, we can see that one cannot necessarily reasonably expect 100% agreement to be achievable with Bach's specific annotations given such variabilities, begging the question: *How much have these variabilities actually affected the results?* It is impossible to know with the information I have, but future research of quantitative studies on this topic will be of great value.

The final insight of this research is about how the generated chord labels could be potentially evaluated. Although this was not the research question concerned in this dissertation, this is still an essential issue and should be addressed in future research. As discussed in Section 4.2.1, the analyses provided by different experts could be significantly different, suggesting that there was no single best answer but rather multiple alternate ones. However, in the existing automatic chord labelling literature, the generated chord labels were still being evaluated using a single ground truth, and the performances were represented by classification accuracy (see Section 3.4.7), which reflected the percentage of exact matches between the generated single labels and the single ground truth. According to Humphrey and Bello (2015) and Koops et al. (2019), this approach could be problematic, because:

- A "wrong" chord label generated by automatic chord labeller could possibly be correct if more alternate ground truths are considered.
- Also, must chord labels be evaluated in a binary way using solely "right" or "wrong"? I.e., could a chord label be partially correct, even if it is different from the single ground truth? Could a distance-based metric (introduced in Section 4.2.2 and Section 4.2.3) instead be potentially used as metric to evaluate the generated chord labels?

Before discussing other alternatives, it is essential to specify the purpose of evaluation. Is the goal about measuring how much can the generated labels mimic a personal annotation style, or reflect a particular, well-defined analytical perspective? If so, the approach of using a single ground truth and accuracy can be reasonable only if the ground truth is contributed exclusively by an individual (such as Koops et al. (2020)) or generated using a well-defined analytical strategy (such as Ju et al. (2019)). However, a more general scenario will be: How is the quality of the generated labels evaluated by a group of experts, in the form of multiple alternate ground truths? In this case, the generated labels will be evaluated in a potentially more general way and won't succumb to any personal style or analytical strategy. Overall, here are my preliminary thoughts on how the generated chord labels could be evaluated in the future research of automatic chord labelling:

- If the ground truth only contains single chord labels, a distance-based metric is worth experimenting, since it reflects a more fine-grained relationship between the generated label and the ground truth than classification accuracy. There are many options, including the ones of chord distance and chord similarity, introduced respectively in Section 4.2.2 and Section 4.2.3.
- Since there are often multiple alternate analyses in chord labelling, a natural way of evaluation is to incorporate multiple parallel tracks of annotations as ground truth, as suggested above. There are a few existing datasets of such kind that already contain multiple chord label annotations (De Clercq and Temperley 2011; Devaney et al. 2015; Koops et al. 2019), and many other alternate evaluation metrics can be explored. For example, could we still use accuracy and consider the generated label correct as long as it matches any of the alternate analyses? If a distance-based metric is used, could the distance be an averaged/weighted value between the generated result and each of the alternate analyses? If the generated results also contain multiple parallel labels, as is the case in Chapter 7, an even wider range of existing metrics from multi-label learning and label distribution learning, or perhaps new metrics such as the inclusive accuracy (proposed in Section 7.2.2), and the ranking accuracy (proposed in Section 7.3.2), can be explored.

Of course, any formal introduction of a new evaluation metric is a demanding task, since it requires justifications of why the old one (accuracy in this case) is inferior, and why the new

one is a better option. Although conducting this research is out of the scope of this dissertation, it is an essential topic that should be discussed in the future research of automatic chord labelling.

8.2 Contributions

A detailed, itemized list of contributions of this research is shown as follows:

- A general discussion of ambiguity, and a discussion of chord labelling ambiguity were introduced in Chapter 1. A survey of methods to address ambiguity in machine learning was introduced in Chapter 2. A brief history of harmony and chord labelling was introduced in Chapter 3, which also contained a survey of automatic chord labelling. Previous discussions of chord labelling ambiguity were introduced in Chapter 4.
- In Chapter 5, an interactive workflow for the semi-automatic chord labelling of Bach chorales was proposed. It combined the consistency of rule-based models with the nuance of manual analysis to generate relatively inexpensive high-quality ground truth. Through a series of experiments on input features, model architectures, and hyper-parameters, different machine learning models for chord labelling were explored. These models could also form an algorithm ensemble to generate chord labels by voting, which improved performance over what could be obtained by the individual algorithms.
- In Chapter 6, a new Bach Chorales Figured Bass (BCFB) dataset was introduced, which consisted of 139 chorales composed by J. S. Bach with the original music and figured bass annotations encoded in MusicXML, **kern, and MEI formats. A comparative study of automatic figured bass annotations of BCFB was also presented, using both rule-based and machine learning approaches. Then, a series of four rule-based algorithms that automatically generated chord labels for Bach chorales based on both figured-bass annotations and the musical surface was implemented. These algorithms were applied to the BCFB dataset, and the resulting parallel chord labels were presented as the new Bach Chorales Multiple Chord Labels (BCMCL) dataset.

- In Chapter 7, multi-label learning and label distribution learning were explored in automatic chord labelling. The parallel chord annotations from BCMCL served as training data for the multi-label learning and label distribution learning, so that the resulting automatic chord labeller could predict one chord label according to a specific labelling perspective, and also multiple alternate labels with membership scores simultaneously.

Overall, this research has made deliberate attempts to consider chord labelling ambiguity and provide various methods for automatic chord labelling using heuristics, single-label learning, multi-label learning, and label distribution learning. Furthermore, two new datasets were made publicly available from this research: The BCFB dataset and the BCMCL dataset. Both have promising applications beyond the scope of this research: The former will facilitate future research revealing insights into Bach's unique compositional style, and potentially provide plausible ideas about his thoughts on counterpoint and harmony. The four separate tracks of chord labels in the latter BCMCL dataset may provide an interesting comparative resource for the empirical study of chord labelling ambiguity: What characteristics of the musical surface and figured bass lead to ambiguity? Specifically, are the number of possible chord labels and the types of chord labels related to some of these characteristics? What are the relationships between the parallel chord labels? Are they in fact similar or different?

8.3 Publicly available resources

The data and code of this research were all made publicly available at Github:

- The rule-based algorithm introduced in Section 5.1 is available at: https://github.com/DDMAL/Flexible_harmonic_chorale_annotations. A newer version of the algorithm is available at: https://github.com/Computational-Cognitive-Musicology-Lab/FlexibleChoraleHarmonicAnalyses, with a detailed documentation on its usage.
- The code of the interactive workflow introduced in Section 5.2 is available at: https://github.com/juyaolongpaul/harmonic_analysis/releases/tag/v1.1.
- The Bach Chorales Figured Bass dataset introduced in Section 6.2 is available at: https://github.com/juyaolongpaul/Bach_chorale_FB/releases/tag/v2.0, also archived at Zenodo: https://zenodo.org/record/5084914#.YOhRZDPivdU.

- The installation guideline for the automatic figured bass annotation algorithm (Section 6.3) is: https://github.com/juyaolongpaul/harmonic_analysis/blob/master/Github_Page/installation_guide_FB.md.

  The installation guideline for the automatic chord labelling algorithm (Section 6.4) is: https://github.com/juyaolongpaul/harmonic_analysis/blob/master/Github_Page/installation_guide_FB_chord.md.

- The Bach Chorales Multiple Chord Labels (BCMCL) dataset introduced in Section 6.5 is available at: https://github.com/juyaolongpaul/BCMCL/releases/tag/v1.0, and BCMCL 1.1 introduced in Section 7.1 is available at: https://github.com/juyaolongpaul/BCMCL/releases/tag/1.1, also archived at Zenodo: https://zenodo.org/record/5068319#.YOFvCRHivdW.

- The installation guideline for multi-label learning and label distribution learning for automatic chord labelling introduced in Section 7.2 and Section 7.3 is available at: https://github.com/juyaolongpaul/harmonic_analysis/blob/master/Github_Page/installation_guide_FB_chord_MLL_LDL.md.

8.4 Limitations and future research

There are also some limitations and compromises associated with this research. One is that the deep learning models used in this research were exclusively fully-connected, feedforward neural networks. Other deep learning models, such as convolutional neural networks, recurrent neural networks, and transformers can also be experimented with. Additionally, all the methods of automatic chord labelling introduced in this research were exclusively applied to Bach chorales, which contained a very specific texture that was largely homorhythmic. Therefore, these methods only had a limited scope of application (e.g., to other chorale music), and they may not apply well to the music with other textures, such as homophony, a superset of homorhythm that includes any music with a primary melodic line accompanied by harmonic support.

Besides the insights introduced in Section 8.1 that offered some possible directions for future research, certain aspects of this research can also be further investigated. In Section 6.3, I introduced automatic figured bass annotation, which could result in a new approach of automatic

chord labelling: First it will be generating figured bass automatically from the music and then converting the figures to chord labels using the rule-based algorithms proposed in Section 6.4; this would allow a chord classifier to take advantage of knowledge implicitly learned from a figured bass classifier during its training. This new approach of automatic chord labelling can be compared against the existing ones presented in this research, which can potentially result in interesting findings and new insights. Also, there are many alternate approaches to multi-label learning and label distribution learning than the ones introduced in Chapter 7, and exploring them can potentially result in better performances for both paradigms. More input features can also be experimented with. For example, information on voice-leading (how one voice moves horizontally) can be added to automatic chord labellers, since this information is essential for human analysts to provide chord label annotations. Pitch spelling is another example, and its information can be added as features as well. However, adding these features will significantly expand the feature vector of automatic chord labellers, and the resulting model might overfit. Therefore, expanding all the datasets used in this research (e.g., the BCFB and the BCMCL datasets) can certainly be helpful too.

Bibliography

Agawu, Kofi. 1994. “Ambiguity in Tonal Music: A Preliminary Study.” *Theory, Analysis and Meaning in Music*, 86–107.

Amershi, Saleema, Maya Cakmak, William Bradley Knox, and Todd Kulesza. 2014. “Power to the People: The Role of Humans in Interactive Machine Learning.” *AI Magazine* 35 (4): 105–20.

Arnold, Franck Thomas. 1931. *The Art of Accompaniment from a Thorough-Bass: As Practised in the XVIIth & XVIIIth Centuries*. Oxford University Press.

Aroyo, Lora, and Chris Welty. 2014. “The Three Sides of Crowdtruth.” *Human Computation* 1 (1).

Awel, Muna Ahmed, and Ali Imam Abidi. 2019. “Review on Optical Character Recognition.” *International Research Journal of Engineering and Technology* 6 (06): 3666–69.

Bach, Carl Philipp Emanuel. 1949. *Essay on the True Art of Playing Keyboard Instruments*. (Berlin, 1753 and 1762). Translated by Williman John Mitchell. 2 vols. New York: W. W. Norton.

Bach, Johann Sebastian, Alfred Dürr, and Werner Neumann. 1954–2007. *Neue Ausgabe Sämtlicher Werke*. Bärenreiter.

Barthélemy, Jerome, and Alain Bonardi. 2001. “Figured Bass and Tonality Recognition.” In *Proceedings of the 2nd International Symposium for Music Information Retrieval Conference*.

Battenberg, Eric, Jitong Chen, Rewon Child, Adam Coates, Yashesh Gaur Yi Li, Hairong Liu, Sanjeev Satheesh, Anuroop Sriram, and Zhenyao Zhu. 2017. “Exploring Neural Transducers for End-to-End Speech Recognition.” In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 206–13. Okinawa: IEEE. https://doi.org/10.1109/ASRU.2017.8268937.

Beach, David W. 1974. “The Origins of Harmonic Analysis.” *Journal of Music Theory* 18 (2): 274–306.

Bent, Ian D., David W. Hughes, Robert C. Provine, Richard Rastall, Anne Kilmer, David Hiley, Janka Szendrei, Thomas B. Payne, Margaret Bent, and Geoffrey Chew. 2001. *Notation*. Oxford University Press. https://doi.org/10.1093/gmo/9781561592630.article.20114.

Benward, Bruce, and Marilyn Saker. 2003. *Music in Theory and Practice*. Vol. 1. McGraw-Hill Higher Education.

Bernstein, David W. 2002. “Nineteenth-Century Harmonic Theory: The Austro-German Legacy.” In *The Cambridge History of Western Music Theory*, edited by Thomas Christensen, 778–811. Cambridge University Press.

Biamonte, Nicole. 2010. “Triadic Modal and Pentatonic Patterns in Rock Music.” *Music Theory Spectrum* 32 (2): 95–110. https://doi.org/10.1525/mts.2010.32.2.95.

Bien, Nicholas, Pranav Rajpurkar, Robyn L. Ball, Jeremy Irvin, Allison Park, Erik Jones, Michael Bereket, et al. 2018. “Deep-Learning-Assisted Diagnosis for Knee Magnetic Resonance Imaging: Development and Retrospective Validation of MRNet.” *PLOS Medicine* 15 (11): e1002699.

Blume, Friedrich. 1928–1940. *Gesamtausgabe Der Musikalischen Werke von Michael Praetorius*. Vol. 1–20. Wolfenbüttel-Berlin: Georg Kallmeyer Verlag.

Böck, Sebastian, Matthew E. P. Davies, and Peter Knees. 2019. “Multi-Task Learning of Tempo and Beat: Learning One to Improve the Other.” In *Proceedings of the 20th International Society of Music Information Retrieval Conference*, 486–93.

Boutell, Matthew R., Jiebo Luo, Xipeng Shen, and Christopher M. Brown. 2004. “Learning Multi-Label Scene Classification.” *Pattern Recognition* 37 (9): 1757–71.

Brinker, Christian, Eneldo Loza Mencía, and Johannes Fürnkranz. 2014. “Graded Multilabel Classification by Pairwise Comparisons.” In *Proceedings of the IEEE International Conference on Data Mining*, 731–36. IEEE.

Burgoyne, John Ashley, Jonathan Wild, and Ichiro Fujinaga. 2011. “An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis.” In *Proceeding of the 12th International Society of Music Information Retrieval Conference*, 633–38.

Carabias-Orti, Julio José, Francisco J. Rodriguez-Serrano, Pedro Vera-Candeas, Nicolás Ruiz-Reyes, and Francisco J. Canadas-Quesada. 2015. “An Audio to Score Alignment Framework Using Spectral Factorization and Dynamic Time Warping.” In *Proceeding of the 16th International Society of Music Information Retrieval Conference*, 742–48.

Chen, Guibin, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. 2017. “Ensemble Application of Convolutional and Recurrent Neural Networks for Multi-Label Text Categorization.” In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2377–83. IEEE.

Chen, Tsung-Ping, and Li Su. 2018. “Functional Harmony Recognition of Symbolic Music Data with Multi-Task Recurrent Neural Networks.” In *Proceedings of 19th International Society for Music Information Retrieval Conference*, 90–97.

Chen, Tsung-Ping, and Li Su. 2019. “Harmony Transformer: Incorporating Chord Segmentation into Harmony Recognition.” In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 259–67.

Cheng, Weiwei, and Eyke Hüllermeier. 2009. “Combining Instance-Based Learning and Logistic Regression for Multilabel Classification.” *Machine Learning* 76: 211–25.

Cheng, Weiwei, Eyke Hüllermeier, and Krzysztof J. Dembczynski. 2010. “Graded Multilabel Classification: The Ordinal Case.” In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 223–30.

Chew, Geoffrey, and revised by Richard Rastall. 2014. *III. History of Western Notation. 4. Mensural Notation from 1500.* Oxford University Press. https://doi.org/10.1093/gmo/9781561592630.013.6002277690. Accessed May 4, 2020.

Choi, Keunwoo, George Fazekas, and Mark Sandler. 2016. “Text-Based LSTM Networks for Automatic Music Composition.” In *Proceedings of the 1st Conference on Computer Simulation of Musical Creativity*, 1–8.

Chuan, Ching-Hua, and Elaine Chew. 2008. “Evaluating and Visualizing Effectiveness of Style Emulation in Musical Accompaniment.” In *Proceedings of 9th International Conference on Music Information Retrieval*, 57–62.

Chuan, Ching-Hua, and Elaine Chew. 2011. “Generating and Evaluating Musical Harmonizations That Emulate Style.” *Computer Music Journal* 35 (4): 64–82.

Clare, Amanda, and Ross D. King. 2001. “Knowledge Discovery in Multi-Label Phenotype Data.” In *Principles of Data Mining and Knowledge Discovery*, edited by Luc De Raedt and Arno Siebes, 42–53. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer.

Cohen, Robin, Mike Schaekermann, Sihao Liu, and Michael Cormier. 2019. “Trusted AI and the Contribution of Trust Modeling in Multiagent Systems.” In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 1644–48.

Cohn, Richard, Brian Hyer, Carl Dahlhaus, Julian Anderson, and Charles Wilson. 2001. *Harmony*. Oxford University Press. https://doi.org/10.1093/gmo/9781561592630.article.50818. Accessed March 27, 2021.

Condit-Schultz, Nathaniel, Yaolong Ju, and Ichiro Fujinaga. 2018. “A Flexible Approach to Automated Harmonic Analysis: Multiple Annotations of Chorales by Bach and Prætorius.” In *Proceedings of the 19th International Society of Music Information Retrieval Conference*, 66–73.

Cortes, Corinna, and Vladimir Vapnik. 1995. “Support-Vector Networks.” *Machine Learning* 20 (3): 273–97.

Cuthbert, Michael Scott, and Christopher Ariza. 2010. “Music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data.” In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, 637–42.

Cuthbert, Michael Scott, Christopher Ariza, and Lisa Friedland. 2011. “Feature Extraction and Machine Learning on Symbolic Music Using the Music21 Toolkit.” In *Proceeding of the 12th International Society of Music Information Retrieval Conference*, 387–92.

Dallora, Ana Luiza, Peter Anderberg, Ola Kvist, Emilia Mendes, Sandra Diaz Ruiz, and Johan Sanmartin Berglund. 2019. “Bone Age Assessment with Various Machine Learning Techniques: A Systematic Literature Review and Meta-Analysis.” *PLOS ONE* 14 (7): e0220242.

David, Hans Theodore, Arthur Mendel, and Christoph Wolff. 1999. *The New Bach Reader: A Life of Johann Sebastian Bach in Letters and Documents*. W. W. Norton & Company.

Dayal, Veneeta. 2004. “The Universal Force of Free Choice.” *Linguistic Variation Yearbook* 4 (1): 5–40.

De Clercq, Trevor. 2015. “A Model for Scale-Degree Reinterpretation: Melodic Structure, Modulation, and Cadence Choice in the Chorale Harmonizations of J. S. Bach.” *Empirical Musicology Review* 10 (3): 188–206.

De Clercq, Trevor, and David Temperley. 2011. “A Corpus Analysis of Rock Harmony.” *Popular Music* 30 (1): 47–70.

De Haas, Willem Bas. 2012. “Music Information Retrieval Based on Tonal Harmony.” Ph.D. Dissertation, Utrecht University.

De Haas, Willem Bas, Frans Wiering, and Remco C. Veltkamp. 2013. “A Geometrical Distance Measure for Determining the Similarity of Musical Harmony.” *International Journal of Multimedia Information Retrieval* 2 (3): 189–202.

Deutsch, Diana, Kevin Dooley, and Trevor Henthorn. 2008. “Pitch Circularity from Tones Comprising Full Harmonic Series.” *The Journal of the Acoustical Society of America* 124 (1): 589–97.

Devaney, Johanna, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula. 2015. “Theme and Variation Encodings with Roman Numerals (TAVERN): A New Data Set for Symbolic Music Analysis.” In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 728–34.

Dietterich, Thomas G. 2000. “Ensemble Methods in Machine Learning.” In *Proceedings of the International Workshop on Multiple Classifier Systems*, 1–15. Springer.

Drabkin, William. 2002. “Heinrich Schenker.” In *The Cambridge History of Western Music Theory*, edited by Thomas Christensen, 812–43. Cambridge University Press.
Eerola, Tuomas, and Petri Toiviainen. 2004. “MIR in Matlab: The MIDI Toolbox.” In *Proceedings of the 5th International Conference on Music Information Retrieval*, 22–27.
Eisen, Michael B., Paul T. Spellman, Patrick O. Brown, and David Botstein. 1998. “Cluster Analysis and Display of Genome-Wide Expression Patterns.” *Proceedings of the National Academy of Sciences* 95 (25): 14863–68.
Elisseeff, André, and Jason Weston. 2002. “A Kernel Method for Multi-Labelled Classification.” In *Advances in Neural Information Processing Systems*, 681–87.
Fails, Jerry Alan, and Dan R. Olsen Jr. 2003. “Interactive Machine Learning.” In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, 39–45.
Flexer, Arthur, and Thomas Grill. 2016. “The Problem of Limited Inter-Rater Agreement in Modelling Music Similarity.” *Journal of New Music Research* 45 (3): 239–51. https://doi.org/10.1080/09298215.2016.1200631.
Flexer, Arthur, and Taric Lallai. 2019. “Can We Increase Inter- and Intra-Rater Agreement in Modeling General Music Similarity?” In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 494–500.
Freedman, Dylan. 2015. “Correlational Harmonic Metrics: Bridging Computational and Human Notions of Musical Harmony.” Bachelor Thesis, Harvard University.
Fuller, Sarah. 2002. “Organum-Discantus-Contrapunctus in the Middle Ages.” In *The Cambridge History of Western Music Theory*, edited by Thomas Christensen, 477–502. Cambridge University Press.
Fürnkranz, Johannes, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. 2008. “Multilabel Classification via Calibrated Label Ranking.” *Machine Learning* 73 (2): 133–53.
Gao, Bin-Bin, Chao Xing, Chen-Wei Xie, Jianxin Wu, and Xin Geng. 2017. “Deep Label Distribution Learning with Label Ambiguity.” *IEEE Transactions on Image Processing* 26 (6): 2825–38.
Geng, Xin. 2016. “Label Distribution Learning.” *IEEE Transactions on Knowledge and Data Engineering* 28 (7): 1734–48.
Ghamrawi, Nadia, and Andrew McCallum. 2005. “Collective Multi-Label Classification.” In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 195–200.
Gibaja, Eva, and Sebastián Ventura. 2015. “A Tutorial on Multilabel Learning.” *ACM Computing Surveys* 47 (3): 1–38. https://doi.org/10.1145/2716262.
Godbole, Shantanu, and Sunita Sarawagi. 2004. “Discriminative Methods for Multi-Labeled Classification.” In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 22–30. Springer.
Granroth-Wilding, Mark Thomas. 2013. “Harmonic Analysis of Music Using Combinatory Categorial Grammar.” Ph.D. Dissertation, University of Edinburgh.
Greff, Klaus, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. 2017. “LSTM: A Search Space Odyssey.” *IEEE Transactions on Neural Networks and Learning Systems* 28 (10): 2222–32. https://doi.org/10.1109/TNNLS.2016.2582924.

Hadjeres, Gaëtan, François Pachet, and Frank Nielsen. 2017. “DeepBach: A Steerable Model for Bach Chorales Generation.” In *Proceedings of the 34th International Conference on Machine Learning*, 1362–71.

Harte, Christopher. 2010. “Towards Automatic Extraction of Harmony Information from Music Signals.” Ph.D. Dissertation, Queen Mary University of London.

Harte, Christopher, Mark Sandler, Samer Abdallah, and Emilia Gómez. 2005. “Symbolic Representation of Musical Chords: A Proposed Syntax for Text Annotations.” In *Proceedings of the 6th International Conference on Music Information Retrieval*, 66–71.

Hedges, Thomas, and Martin Rohrmeier. 2011. “Exploring Rameau and Beyond: A Corpus Study of Root Progression Theories.” In *Proceedings of International Conference on Mathematics and Computation in Music*, 334–37. Springer.

Heim, Irene. 2002. “File Change Semantics and the Familiarity Theory of Definiteness.” *Formal Semantics: The Essential Readings*, 223–48.

Hoffman, Tim, and William P. Birmingham. 2000. “A Constraint Satisfaction Approach to Tonal Harmonic Analysis.” Technical Report. Electrical Engineering and Computer Science Department. CSE-TR-397-99. University of Michigan.

Holtmeier, Ludwig. 2007. “Heinichen, Rameau, and the Italian Thoroughbass Tradition: Concepts of Tonality and Chord in the Rule of the Octave.” *Journal of Music Theory* 51 (1): 5–49. https://doi.org/10.1215/00222909-2008-022.

Humphrey, Eric, and Juan Pablo Bello. 2015. “Four Timely Insights on Automatic Chord Estimation.” In *Proceedings of the 16th International Society for Music Information Retrieval Conference*, 673–79.

Huron, David. 1999. *Music Research Using Humdrum: A User’s Guide*.

Huron, David. 2002. “Music Information Processing Using the Humdrum Toolkit: Concepts, Examples, and Lessons.” *Computer Music Journal* 26 (2): 11–26.

Hyer, Brian. 2011. “What Is a Function?” In *The Oxford Handbook of Neo-Riemannian Music Theories*, 92–139. Oxford University Press.

Illescas, Plácido R., David Rizo, and José Manuel Iñesta. 2007. “Harmonic, Melodic, and Functional Automatic Analysis.” In *Proceedings of International Computer Music Conference*, 165–68.

Illescas, Plácido R., David Rizo, José Manuel Iñesta, and Rafael Ramírez. 2011. “Learning Melodic Analysis Rules.” In *Proceedings of the International Workshop on Music and Machine Learning*.

Inman, Samantha M. 2018. “Introduction to Graduate Theory: Teaching Tonal Hierarchy through Bach.” *Bach* 49 (2): 345–64. https://doi.org/10.22513/bach.49.2.0345.

Ivakhnenko, Alekseui Grigorevich, and Valentin G. Lapa. 1965. *Cybernetic Predicting Devices*. New York: CCM Information Corporation.

Jain, Himanshu, Yashoteja Prabhu, and Manik Varma. 2016. “Extreme Multi-Label Loss Functions for Recommendation, Tagging, Ranking & Other Missing Label Applications.” In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 935–44.

Ji, Shuiwang, Lei Tang, Shipeng Yu, and Jieping Ye. 2008. “Extracting Shared Subspace for Multi-Label Classification.” In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 381–89.

Ju, Yaolong, Nathaniel Condit-Schultz, Claire Arthur, and Ichiro Fujinaga. 2017. “Non-Chord Tone Identification Using Deep Neural Networks.” In *Proceedings of the 4th International Workshop on Digital Libraries for Musicology - DLfM ’17*, 13–16. Shanghai, China: ACM Press. https://doi.org/10.1145/3144749.3144753.Ju, Yaolong, Samuel Howes, Cory McKay, Nathaniel Condit-Schultz, Jorge Calvo-Zaragoza, and Ichiro Fujinaga. 2019. “An Interactive Workflow for Generating Chord Labels for Homorhythmic Music in Symbolic Formats.” In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 862–69.

Ju, Yaolong, Sylvain Margot, Cory McKay, Luke Dahn, and Ichiro Fujinaga. 2020. “Automatic Figured Bass Annotation Using the New Bach Chorales Figured Bass Dataset.” In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 640–46.

Ju, Yaolong, Sylvain Margot, Cory McKay, and Ichiro Fujinaga. 2020a. “Automatic Chord Labelling: A Figured Bass Approach.” In *Proceedings of the 7th International Conference on Digital Libraries for Musicology*, 27–31.

Ju, Yaolong, Sylvain Margot, Cory McKay, and Ichiro Fujinaga. 2020b. “Figured Bass Encodings for Bach Chorales in Various Symbolic Formats: A Case Study.” In *Proceedings of the Music Encoding Conference*, 71–73.

Jukić, Samed, Dino Kečo, and Jasmin Kevrić. 2018. “Majority Vote of Ensemble Machine Learning Methods for Real-Time Epilepsy Prediction Applied on EEG Pediatric Data.” *TEM Journal* 7 (2): 313–18.

Katakis, Ioannis, Grigorios Tsoumakas, and Ioannis Vlahavas. 2008. “Multilabel Text Classification for Automated Tag Suggestion.” In *Proceedings of the ECML/PKDD Discovery Chanllenge*, 75–83.

Kim, Yoon. 2014. “Convolutional Neural Networks for Sentence Classification.” In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–51.

Klapuri, Anssi. 2006. “Introduction to Music Transcription.” In *Signal Processing Methods for Music Transcription*, 3–20. Springer.

Koelsch, Stefan, Elisabeth Kasper, Daniela Sammler, Katrin Schulze, Thomas Gunter, and Angela D. Friederici. 2004. “Music, Language and Meaning: Brain Signatures of Semantic Processing.” *Nature Neuroscience* 7 (3): 302–7.

Koops, Hendrik Vincent. 2019. “Computational Modelling of Variance in Musical Harmony.” Ph.D. Dissertation, Utrecht University.

Koops, Hendrik Vincent, Willem Bas De Haas, Jeroen Bransen, and Anja Volk. 2020. “Automatic Chord Label Personalization through Deep Learning of Shared Harmonic Interval Profiles.” *Neural Computing and Applications* 32 (4): 929–39. https://doi.org/10.1007/s00521-018-3703-y.

Koops, Hendrik Vincent, Willem Bas De Haas, John Ashley Burgoyne, Jeroen Bransen, Anna Kent-Muller, and Anja Volk. 2019. “Annotator Subjectivity in Harmony Annotations of Popular Music.” *Journal of New Music Research* 48 (3): 232–52.

Kosta, Stefan. 2000. *Tonal Harmony: With an Introduction to Twentieth-Century Music*. McGraw-Hill.

Kostka, Stefan, and Dorothy Payne. 1995. *Workbook for Tonal Harmony*. McGraw Hill.

Kröger, Pedro, Alexandre Passos, and Marcos Sampaio. 2010. “A Survey of Automated Harmonic Analysis Techniques.” Technical report. Universidade Federal da Bahia.
Kröger, Pedro, Alexandre Passos, Marcos Sampaio, and Givaldo De Cidra. 2008. “Rameau: A System for Automatic Harmonic Analysis.” In *Proceedings of International Computer Music Conference*, 273–81.
Krumhansl, Carol L. 1990. *Cognitive Foundations of Musical Pitch*. Oxford University Press.
Laitz, Steven Geoffrey. 2012. *The Complete Musician: An Integrated Approach to Tonal Theory, Analysis, and Listening*. Oxford University Press.
Lerdahl, Fred. 2004. *Tonal Pitch Space*. Oxford University Press.
Lerdahl, Fred, and Ray Jackendoff. 1983. *Toward a Generative Theory of Tonal Music*. MIT Press.
Lester, Joel. 1994. *Compositional Theory in the Eighteenth Century*. Harvard University Press.
Lester, Joel. 2002. “Rameau and Eighteenth-Century Harmonic Theory.” In *The Cambridge History of Western Music Theory*, edited by Thomas Christensen, 753–77. Cambridge University Press.
Lin, Hsuan-Tien, Chih-Jen Lin, and Ruby C. Weng. 2007. “A Note on Platt’s Probabilistic Outputs for Support Vector Machines.” *Machine Learning* 68 (3): 267–76.
López, Néstor Nápoles. 2017. “Automatic Harmonic Analysis of Classical String Quartets from Symbolic Score.” Master’s Thesis, Universitat Pompeu Fabra.
MacDonald, Maryellen C. 1993. “The Interaction of Lexical and Syntactic Ambiguity.” *Journal of Memory and Language* 32 (5): 692–715.
MacDonald, Maryellen C., Neal J. Pearlmutter, and Mark S. Seidenberg. 1994. “The Lexical Nature of Syntactic Ambiguity Resolution.” *Psychological Review* 101 (4): 676.
Manam, Chaithanya, and Alexander Quinn. 2018. “WingIt: Efficient Refinement of Unclear Task Instructions.” In *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, 108–16.
Masada, Kristen, and Razvan Bunescu. 2017. “Chord Recognition in Symbolic Music Using Semi-Markov Conditional Random Fields.” In *Proceedings of 18th International Society for Music Information Retrieval Conference*, 272–78.
Masada, Kristen, and Razvan Bunescu. 2019. “Chord Recognition in Symbolic Music: A Segmental CRF Model, Segment-Level Features, and Comparative Evaluations on Classical and Popular Music.” *Transactions of the International Society for Music Information Retrieval* 2 (1): 1–13.
Maxwell, H. John. 1992. “An Expert System for Harmonic Analysis of Tonal Music.” In *Understanding Music With AI: Perspectives on Music Cognition*, 335–53. MIT Press, Cambridge, MA.
McCallum, Andrew Kachites. 1999. “Multi-Label Text Classification with a Mixture Model Trained by EM.” In *Proceedings of the AAAI Workshop on Text Learning*.
McKay, Cory, Julie Cumming, and Ichiro Fujinaga. 2018. “JSymbolic 2.2: Extracting Features from Symbolic Music for Use in Musicological and MIR Research.” In *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 348–54. Paris, France: ISMIR.
McVicar, Matt, Raul Santos-Rodriguez, Yizhao Ni, and Tijl De Bie. 2014. “Automatic Chord Estimation from Audio: A Review of the State of the Art.” *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22 (2): 556–75. https://doi.org/10.1109/TASLP.2013.2294580.

Mearns, Lesley. 2013. “The Computational Analysis of Harmony in Western Art Music.” Ph.D. Dissertation, Queen Mary University of London.
Micchi, Gianluca, Mark Gotham, and Mathieu Giraud. 2020. “Not All Roads Lead to Rome: Pitch Representation and Model Architecture for Automatic Harmonic Analysis.” *Transactions of the International Society for Music Information Retrieval* 3 (1): 42–54.
Moss, Fabian C., Markus Neuwirth, Daniel Harasim, and Martin Rohrmeier. 2019. “Statistical Characteristics of Tonal Harmony: A Corpus Study of Beethoven’s String Quartets.” Edited by Carla M.A. Pinto. *PLOS ONE* 14 (6): e0217242. https://doi.org/10.1371/journal.pone.0217242.
Mouton, Rémy, and François Pachet. 1995. “The Symbolic vs. Numeric Controversy in Automatic Analysis of Music.” In *Proceedings of IJCAI’95 Workshop on Artificial Intelligence and Music*, 32–40.
Mumpower, Jeryl L., and Thomas R. Stewart. 1996. “Expert Judgement and Expert Disagreement.” *Thinking & Reasoning* 2 (2–3): 191–212.
Nam, Jinseok, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. “Large-Scale Multi-Label Text Classification—Revisiting Neural Networks.” In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 437–52. Springer.
Nápoles López, Néstor, Gabriel Vigliensoni, and Ichiro Fujinaga. 2018. “Encoding Matters.” In *Proceedings of the 5th International Conference on Digital Libraries for Musicology - DLfM ’18*, 69–73. Paris, France: ACM Press. https://doi.org/10.1145/3273024.3273027.
Nápoles López, Néstor, Gabriel Vigliensoni, and Ichiro Fujinaga. 2019. “The Effects of Translation Between Symbolic Music Formats: A Case Study with Humdrum, Lilypond, MEI, and MusicXML.” In *Music Encoding Conference*.
Neuwirth, Markus, Daniel Harasim, Fabian Claude Moss, and Martin Rohrmeier. 2018. “The Annotated Beethoven Corpus (ABC): A Dataset of Harmonic Analyses of All Beethoven String Quartets.” *Frontiers in Digital Humanities* 5: 16.
Ni, Yizhao, Matt McVicar, Raul Santos-Rodriguez, and Tijl De Bie. 2013. “Understanding Effects of Subjectivity in Measuring Chord Estimation Accuracy.” *IEEE Transactions on Audio, Speech, and Language Processing* 21 (12): 2607–15.
Nicholls, Michael E. R., Owen Churches, and Tobias Loetscher. 2018. “Perception of an Ambiguous Figure Is Affected by Own-Age Social Biases.” *Scientific Reports* 8 (1): 12661. https://doi.org/10.1038/s41598-018-31129-7.
Normann, Immanuel, Hendrik Purwins, and Klaus Obermayer. 2001. “Spectrum of Pitch Differences Models the Perception of Octave Ambiguous Tones.” In *Proceeding of International Computer Music Conference*, 274–76.
Pachet, Francois. 1991. “A Meta-Level Architecture for the Analysis of Jazz Chord Sequences.” In *Proceedings of International Computer Music Conference*, 266–69.
Pardo, Bryan, and William P. Birmingham. 2002. “Algorithms for Chordal Analysis.” *Computer Music Journal* 26 (2): 27–49.
Patel, Aniruddh D. 2010. *Music, Language, and the Brain*. Oxford University Press, USA.
Pernet, Cyril. 2015. “Null Hypothesis Significance Testing: A Short Tutorial.” *F1000Research* 4. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5635437/. Accessed March 13, 2021.
Pickens, Jeremy. 2004. “Harmonic Modeling for Polyphonic Music Retrieval.” Ph.D. Dissertation, University of Massachusetts at Amherst.

Prabhavalkar, Rohit, Kanishka Rao, Tara N. Sainath, Bo Li, Leif Johnson, and Navdeep Jaitly. 2017. “A Comparison of Sequence-to-Sequence Models for Speech Recognition.” In *Proceedings of Interspeech*, 939–43. https://doi.org/10.21437/Interspeech.2017-233.

Qi, Guo-Jun, Xian-Sheng Hua, Yong Rui, Jinhui Tang, Tao Mei, and Hong-Jiang Zhang. 2007. “Correlative Multi-Label Video Annotation.” In *Proceedings of the 15th ACM International Conference on Multimedia*, 17–26.

Quinn, Ian. 2010. “Are Pitch-Class Profiles Really Key for Key.” *Zeitschrift Der Gesellschaft Der Musiktheorie* 7: 151–63.

Quinn, Ian, and Panayotis Mavromatis. 2011. “Voice-Leading Prototypes and Harmonic Function in Two Chorale Corpora.” In *International Conference on Mathematics and Computation in Music*, 230–40. Springer.

Quinn, Ian, and Christopher White. 2013. “Expanding Notions of Harmonic Function through a Corpus Analysis of the Bach Chorales.” In *Joint Meeting of the Society for Music Theory and the American Musicological Society*.

Räbiger, Stefan, Gizem Gezici, Yücel Saygın, and Myra Spiliopoulou. 2018. “Predicting Worker Disagreement for More Effective Crowd Labeling.” In *Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*, 179–88. IEEE.

Radicioni, Daniele P., and Roberto Esposito. 2007. “Tonal Harmony Analysis: A Supervised Sequential Learning Approach.” In *Proceedings of Artificial Intelligence and Human-Oriented Computing*, edited by Roberto Basili and Maria Teresa Pazienza, 638–49. Berlin, Heidelberg: Springer.

Radicioni, Daniele P., and Roberto Esposito. 2010. “BREVE: An HMPerceptron-Based Chord Recognition System.” In *Advances in Music Information Retrieval*, edited by Z. W. Raś and A. A. Wieczorkowska, 143–64. Springer.

Rajpurkar, Pranav, Jeremy Irvin, Robyn L. Ball, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, and Curtis P. Langlotz. 2018. “Deep Learning for Chest Radiograph Diagnosis: A Retrospective Comparison of the CheXNeXt Algorithm to Practicing Radiologists.” *PLoS Medicine* 15 (11): e1002686.

Raphael, Christopher, and Joshua Stoddard. 2004. “Functional Harmonic Analysis Using Probabilistic Models.” *Computer Music Journal* 28 (3): 45–52.

Read, Jesse, Bernhard Pfahringer, and Geoff Holmes. 2008. “Multi-Label Classification Using Ensembles of Pruned Sets.” In *Proceedings of the International Conference on Data Mining*, 995–1000. IEEE.

Read, Jesse, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. “Classifier Chains for Multi-Label Classification.” *Machine Learning* 85 (3): 333–59.

Remeš, Derek. 2019. *Realizing Thoroughbass Chorales in the Circle of J. S. Bach (2 Vols.)*. Wayne Leupold Editions.

Riemann, Hugo. 1896. *Harmony Simplified; Or, The Theory of the Tonal Functions of Chords*. Translated by Henry Bewerunge. Augener.

Rizo, David, Plácido R. Illescas, and José Manuel Iñesta. 2016. “Interactive Melodic Analysis.” In *Computational Music Analysis*, edited by David Meredith, 191–219. Springer International Publishing.

Rohrmeier, Martin. 2006. “Towards Modelling Harmonic Movement in Music: Analysing Properties and Dynamic Aspects of PC Set Sequences in Bach’s Chorales.” Technical Report DCRR-004, Darwin College, University of Cambridge.

Rohrmeier, Martin. 2011. “Towards a Generative Syntax of Tonal Harmony.” *Journal of Mathematics and Music* 5 (1): 35–53.
Rohrmeier, Martin, and Ian Cross. 2008. “Statistical Properties of Tonal Harmony in Bach’s Chorales.” In *Proceedings of the 10th International Conference on Music Perception and Cognition*, 619–27. Hokkaido University Sapporo, Japan.
Ruamviboonsuk, Paisan, Jonathan Krause, Peranut Chotcomwongse, Rory Sayres, Rajiv Raman, Kasumi Widner, Bilson J. L. Campana, et al. 2019. “Deep Learning versus Human Graders for Classifying Diabetic Retinopathy Severity in a Nationwide Screening Program.” *Digital Medicine* 2 (1): 1–9.
Salamon, Justin, and Juan Pablo Bello. 2017. “Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification.” *IEEE Signal Processing Letters* 24 (3): 279–83.
Sanden, Chris, and John Z. Zhang. 2011. “Enhancing Multi-Label Music Genre Classification through Ensemble Techniques.” In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 705–14.
Sapp, Craig Stuart. 2007. “Computational Chord-Root Identification in Symbolic Musical Data: Rationale, Methods, and Applications.” *Computing in Musicology* 15: 99–119.
Saslaw, Janna K. 1992. “Gottfried Weber and the Concept of Mehrdeutigkeit.” Ph.D. Dissertation, Columbia University.
Sayres, Rory, Ankur Taly, Ehsan Rahimy, Katy Blumer, David Coz, Naama Hammel, Jonathan Krause, et al. 2019. “Using a Deep Learning Algorithm and Integrated Gradients Explanation to Assist Grading for Diabetic Retinopathy.” *Ophthalmology* 126 (4): 552–64.
Schaekermann, Mike, Graeme Beaton, Minahz Habib, Andrew Lim, Kate Larson, and Edith Law. 2019. “Understanding Expert Disagreement in Medical Data Analysis through Structured Adjudication.” *Proceedings of the ACM on Human-Computer Interaction* 3 (CSCW): 1–23. https://doi.org/10.1145/3359178.
Schaekermann, Mike, Graeme Beaton, Elaheh Sanoubari, Andrew Lim, Kate Larson, and Edith Law. 2020. “Ambiguity-Aware AI Assistants for Medical Data Analysis.” In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 1–14. ACM. https://doi.org/10.1145/3313831.3376506.
Schapire, Robert E., and Yoram Singer. 2000. “BoosTexter: A Boosting-Based System for Text Categorization.” *Machine Learning* 39 (2): 135–68.
Scholz, Ricardo, Vitor Dantas, and Geber Ramalho. 2005. “Automating Functional Harmonic Analysis: The Funchal System.” In *Proceedings of the Seventh IEEE International Symposium on Multimedia*, 6–12.
Seide, Frank, Gang Li, and Dong Yu. 2011. “Conversational Speech Transcription Using Context-Dependent Deep Neural Networks.” In *Proceedings of Interspeech*, 437–40.
Selway, Anna, Hendrik Vincent Koops, Anja Volk, David Bretherton, Nicholas Gibbins, and Richard Polfreman. 2020. “Explaining Harmonic Inter-Annotator Disagreement Using Hugo Riemann’s Theory of ‘Harmonic Function.’” *Journal of New Music Research* 49 (2): 136–50.
Serra, Joan, Meinard Müller, Peter Grosche, and Josep Arcos. 2014. “Unsupervised Music Structure Annotation by Time Series Structure Features and Segment Similarity.” *IEEE Transactions on Multimedia* 16 (5): 1229–40.

Shepard, Roger N. 1964. “Circularity in Judgments of Relative Pitch.” *The Journal of the Acoustical Society of America* 36 (12): 2346–53.
Shibutani, Takayuki, Kenichi Nakajima, Hiroshi Wakabayashi, Hiroshi Mori, Shinro Matsuo, Hiroto Yoneyama, Takahiro Konishi, Koichi Okuda, Masahisa Onoguchi, and Seigo Kinuya. 2019. “Accuracy of an Artificial Neural Network for Detecting a Regional Abnormality in Myocardial Perfusion SPECT.” *Annals of Nuclear Medicine* 33 (2): 86–92.
Shrivaslava, Harsh, Yfang Yin, Rajiv Ratn Shah, and Roger Zimmermann. 2020. “MT-GCN for Multi-Label Audio Tagging with Noisy Labels.” In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 136–40.
Smith, Jordan, and Elaine Chew. 2013. “A Meta-Analysis of the MIREX Structure Segmentation Task.” In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 251–56.
Sun, Chen, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. “Revisiting Unreasonable Effectiveness of Data in Deep Learning Era.” In *Proceedings of the IEEE International Conference on Computer Vision*, 843–52.
Taube, Heinrich. 1999. “Automatic Tonal Analysis: Toward the Implementation of a Music Theory Workbench.” *Computer Music Journal* 23 (4): 18–32.
Team, R. Core. 2013. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna. https://www.r-project.org/. Accessed October 19, 2020.
Temperley, David. 1997. “An Algorithm for Harmonic Analysis.” *Music Perception* 15 (1): 31–68.
Temperley, David. 2001. *The Cognition of Basic Musical Structures*. MIT Press.
Temperley, David, and Daniel Sleator. 1999. “Modeling Meter and Harmony: A Preference-Rule Approach.” *Computer Music Journal* 23 (1): 10–27.
Tojo, Satoshi, Yoshinori Oka, and Masafumi Nishida. 2006. “Analysis of Chord Progression by HPSG.” In *Proceedings of the International Conference on Artificial Intelligence and Applications*, 305–10.
Trohidis, Konstantinos, Grigorios Tsoumakas, George Kalliris, and Ioannis P. Vlahavas. 2008. “Multi-Label Classification of Music into Emotions.” In *Proceedings of the 9th International Conference on Music Information Retrieval*, 325–30.
Tsoumakas, Grigorios, and Ioannis Katakis. 2007. “Multi-Label Classification: An Overview.” *International Journal of Data Warehousing and Mining (IJDWM)* 3 (3): 1–13.
Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. 2010. “Random K-Labelsets for Multilabel Classification.” *IEEE Transactions on Knowledge and Data Engineering* 23 (7): 1079–89.
Tsoumakas, Grigorios, and Ioannis Vlahavas. 2007. “Random K-Labelsets: An Ensemble Method for Multilabel Classification.” In *Proceedings of the European Conference on Machine Learning*, 406–17. Springer.
Tsui, Wan Shun Vincent. 2002. “Harmonic Analysis Using Neural Networks.” Master’s Thesis, University of Toronto.
Tymoczko, Dmitri, Mark Gotham, Michael Scott Cuthbert, and Christopher Ariza. 2019. “The Romantext Format: A Flexible and Standard Method for Representing Roman Numeral Analyses.” In *Proceedings of the 20th International Society for Music Information Retrieval Conference*, 123–29.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." In *Advances in Neural Information Processing Systems*, edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 5998–6008.

Vogler, Georg Joseph. 1776. *Gründe Der Kuhrpfälzischen Tonschule in Beispielen*. Offenbach am Main: J. André.

Voigt, Daniel, Michael Döllinger, Anxiong Yang, Ulrich Eysholdt, and Jörg Lohscheller. 2010. "Automatic Diagnosis of Vocal Fold Paresis by Employing Phonovibrogram Features and Machine Learning Methods." *Computer Methods and Programs in Biomedicine* 99 (3): 275–88.

Warby, Simon, Sabrina Wendt, Peter Welinder, Emil Munk, Oscar Carrillo, Helge Sorensen, Poul Jennum, Paul Peppard, Pietro Perona, and Emmanuel Mignot. 2014. "Sleep-Spindle Detection: Crowdsourcing and Evaluating Performance of Experts, Non-Experts and Automated Methods." *Nature Methods* 11 (4): 385–92.

Wead, Adam, and Ian Knopke. 2007. "A Computer-Based Implementation of Basso Continuo Rules for Figured Bass Realizations." In *Proceedings of International Computer Music Conference*, 188–91.

Williams, Peter, and David Ledbetter. 2001. "Figured Bass." In *Oxford Music Online*. Oxford University Press. https://doi.org/10.1093/gmo/9781561592630.article.09623. Accessed March 4, 2020.

Willingham, Timothy Judson. 2013. "The Harmonic Implications of the Non-Harmonic Tones in the Four-Part Chorales of Johann Sebastian Bach." Ph.D. Dissertation, Liberty University.

Wimmer, Marina C., Martin J. Doherty, and W. Andrew Collins. 2011. "The Development of Ambiguous Figure Perception." *Monographs of the Society for Research in Child Development* 76 (1): 1–130.

Winograd, Terry. 1968. "Linguistics and the Computer Analysis of Tonal Harmony." *Journal of Music Theory* 12 (1): 2–49.

Woolhouse, Matthew. 2015. "Probability and Style in the Chorales of J. S. Bach." *Empirical Musicology Review* 10 (3): 207–14.

Wu, Ting-Fan, Chih-Jen Lin, and Ruby C. Weng. 2004. "Probability Estimates for Multi-Class Classification by Pairwise Coupling." *Journal of Machine Learning Research* 5 (Aug): 975–1005.

Yan, Rong, Jelena Tesic, and John R. Smith. 2007. "Model-Shared Subspace Boosting for Multi-Label Classification." In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 834–43.

Yang, Pengcheng, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. "SGM: Sequence Generation Model for Multi-Label Classification." In *Proceedings of the 27th International Conference on Computational Linguistics*, 3915–26.

Yang, Yiming, and Xin Liu. 1999. "A Re-Examination of Text Categorization Methods." In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 42–49.

Yeary, Mark. 2011. "Perception, Pitch, and Musical Chords." Ph.D. Dissertation, University of Chicago.

Yin, Lijun, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J. Rosato. 2006. “A 3D Facial Expression Database for Facial Behavior Research.” In *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, 211–16. IEEE.

Zhang, Min-Ling, and Kun Zhang. 2010. “Multi-Label Learning by Exploiting Label Dependency.” In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 999–1008.

Zhang, Min-Ling, and Zhi-Hua Zhou. 2006. “Multilabel Neural Networks with Applications to Functional Genomics and Text Categorization.” *IEEE Transactions on Knowledge and Data Engineering* 18 (10): 1338–51.

Zhang, Min-Ling, and Zhi-Hua Zhou. 2007. “ML-KNN: A Lazy Learning Approach to Multi-Label Learning.” *Pattern Recognition* 40 (7): 2038–48.

Zhang, Min-Ling, and Zhi-Hua Zhou. 2014. “A Review on Multi-Label Learning Algorithms.” *IEEE Transactions on Knowledge and Data Engineering* 26 (8): 1819–37.

Zhao, Fang, Yongzhen Huang, Liang Wang, and Tieniu Tan. 2015. “Deep Semantic Ranking Based Hashing for Multi-Label Image Retrieval.” In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1556–64.

Zhao, Feng, and Xianghua Xie. 2013. “An Overview on Interactive Medical Image Segmentation.” *Annals of the BMVA* 2013 (7): 1–22.

Zheng, Renjie, Mingbo Ma, and Liang Huang. 2018. “Multi-Reference Training with Pseudo-References for Neural Translation and Text Generation.” In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3188–97. Brussels, Belgium: Association for Computational Linguistics.

Zhou, Zhi-Hua. 2012. *Ensemble Methods: Foundations and Algorithms*. CRC press.

Zhu, Shenghuo, Xiang Ji, Wei Xu, and Yihong Gong. 2005. “Multi-Labelled Classification Using Maximum Entropy Method.” In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 274–81.