An Interactive System for Guitar Chord Practice with a Data-driven Approach

(データドリブンな手法を用いた
インタラクティブなギターコード練習システム)

37-156432　Shunya Ariga (有賀 竣哉)

Supervisor: Assoc. Prof. Koji Yatani (矢谷 浩司)

Department of Electrical Engineering and Information Systems
Graduate School of Engineering
The University of Tokyo

3 February 2017

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Koji Yatani, for insightful comments and advice throughout my master course. He gave me a chance to consider what I want to do in my master course and allowed me to take on a musical topic as a thesis theme, which is fascinating to me. This thesis would not be possible without his generous support.

Furthermore, I am deeply grateful to Masataka Goto at Advanced Industrial Science and Technology (AIST). He gave me thoughtful advice and feedback on musical signal processing as well as my research. He also gave me an opportunity to take part in a summer internship of his research group. I led a project about automatic arrangement for guitar solo performance in the internship. In my internship project, Satoru Fukayama at AIST mainly guided me from the very fundamental concept in the research area of Musical Information Retrieval. I here express my sincere appreciation for his kind coaching.

The members of our laboratory have vastly contributed to my thesis. Takahiro Hashizume eagerly helped me when I was struggling to write my first paper to international conference. Namiki Shimoo gave me ideas and cooperated with my project as an experienced guitar player. I also would like to thank all the participants in my evaluation, who willingly shared their time for my five-day long experiments.

Finally, I greatly thank my parents and brother, who always supported me throughout my graduate school life. I am indebted to my parents for giving me the opportunity to study at the Master program at the university.

Abstract

Musical instrument playing is a skill many people desire to acquire, and learners now have a wide variety of learning materials. However, their volume is enormous, and novice learners may easily get lost in which songs they should practice first. My analyses on the Billboard music dataset revealed that occurrences of chords and their transitions follow the power law, confirming their skewed distributions. This leads me to take a data-driven approach to identifying important and easy-to-learn chords and songs. I develop novel metrics for considering both chord frequency and difficulty, and design an interactive system for guitar practice, called Strummer. In Strummer, learners start with songs containing chords that are frequently used and easy to play. This practice design is intended to encourage smooth skill transfers to songs that learners even have not seen. A quantitative examination confirms that this approach can support skill transfer to a larger set of songs than the naïve method bases on the sum of chord difficulty. I also conducted a comparative user study, uncovering benefits and future improvements of the data-driven approach and Strummer. I conclude this thesis with discussing design implications learned from the user study and future research directions.

概要

楽器の演奏は多くの人が望むスキルであり，様々な練習方法や教材が考案されているが，その種類の多さゆえ初心者は何を練習すればよいのか迷いがちである．さらに独学では自分の演奏の可否を判断して修正することが難しく，これらはモチベーション低下につながる問題となっている．本稿では，データドリブンでインタラクティブなギターコード練習システム「Strummer」を提案する．Strummer システムの開発にあたり，コードラベルが付与された 727 曲分の楽曲データを解析し，コードの出現率と難易度を考慮した曲の重要性の尺度を新たに定義した．これをシステム中で用いることで，ユーザーは簡単かつ重要な楽曲から練習が可能になる．さらに，本システムは音響信号解析によってユーザーが正しいコードを弾いたのかを認識してフィードバックする機能を有しており，インタラクティブかつ段階的な練習を提供する．Strummer の提供する楽曲のコード演奏練習は，スムーズに未知の曲でも弾けるようにすることを念頭に置いて設計されている．データセットを用いた解析により，コードの難易度総和に準じた楽曲提示手法と比較して，新たに提案した楽曲提示手法がより多くの楽曲で使われるコードをカバーできることを示した．さらに，上述の 2 種の楽曲提示手法で比較実験を行い，両者で練習した実験参加者が感じた負担に有意差が無いという結果を得たため，提案する楽曲提示手法が有利である根拠となった．

Table of contents

List of figures viii

List of tables ix

1 Introduction 1

1.1 Background 1

1.2 Contributions 2

2 Related Work 4

2.1 Music Practice Support 4

2.1.1 Piano 4

2.1.2 Guitar 5

2.1.3 Other Instruments 6

2.1.4 Music Game Applications 7

2.1.5 Summary 7

2.2 Music Performance Support 8

2.3 Pedagogy and Skill Development 8

2.3.1 Classical Pedagogy and Conventional Textbooks 8

2.3.2 Dalcroze method 9

2.3.3 Dreyfus's Skill Development 9

2.3.4 Summary 9

2.4 Music Information Retrieval 10

3 Chord Dataset 11

3.1 Dataset Overview 11

3.2 Terminology 12

3.3 Genre Determination 14
3.4 Dataset Parsing and Chord Simplification 15
3.4.1 Representation Simplification 15
3.4.2 Component Simplification 16
3.5 Analysis Results 17
3.6 Chord and Transition Difficulty Rating 18
3.6.1 Chord (Uni-chord) Difficulty Rating 18
3.6.2 Chord Transition (Bi-chord) Difficulty Rating 20
3.6.3 Chord Transition Difficulty Model 21

4 Chord and Song Prioritization Algorithms 24
4.1 Chord Primariness 24
4.2 Song Primariness 26
4.3 Song Primariness Quantitative Evaluation 27
4.4 Shortest Chord Path 28

5 The Strummer System 30
5.1 Strummer Tutorial Interface 30
5.1.1 Listen Stage 30
5.1.2 Chord Practice Stage 31
5.1.3 Chord Transition Practice Stage 34
5.1.4 Segment Practice Stage 36
5.1.5 Rehearsal Stage 37
5.1.6 Live Concert Stage 37
5.2 Implementation 38
5.3 Performance Assessment Module 38

6 User Evaluation 41
6.1 Objectives of the User Study 41
6.2 Participants 42
6.3 Practiced Songs 42
6.4 Procedure 42
6.5 Evaluation Results 45

6.5.1 Sight Reading Task Performance 45
6.5.2 Questionnaire 46
6.5.3 Qualitative Results 48

7 Discussion 52
7.1 Effects of Prioritization Algorithms 52
7.2 Interface and Tutorial Design 52
7.3 Audio Analysis and Performance Assessment 53
7.4 Limitations and Future Improvements 54

8 Conclusion 57

List of publications 58

References 59

Appendix A Guitar Chords Diagrams and Difficulties 62

List of figures

3.1 An example of chord notation in the Billboard dataset. 12
3.2 An description of annotation data. 13
3.3 An example of the converted JSON file. 18
3.4 Log-log plots of uni-chord occurrences. 19
3.5 Log-log plots of bi-chord occurrences. 19
3.6 A description of finger positions in non-barre and barre chord (*C* to *Gm*). . . 21

4.1 The map of representative chords in the *CP* space. 25
4.2 Plots of k%-playable songs (genre: pops) across the number of songs completed ($k = 80$ and 100) in two ranking algorithms (*SP* and *BL*). 28

5.1 Screenshots of the Strummer system interface. 31
5.2 An overview of chord practice stage workflow. 32
5.3 An example of chord stage top menu which include mastered chords. 34
5.4 An overview of chord transition practice stage workflow. 35
5.5 Screenshots of the Strummer system interface. 36
5.6 An overview of the performance assessment workflow. 40

6.1 A scene of sight reading task in the user study. 44
6.2 Quantitative results. 46
6.3 All results of the questionnaire responses. 47

7.1 The total chord difficulty shift in the SP and baseline conditions. 53

A.1 All chord diagrams of root *Ab* and *A*. 63
A.2 All chord diagrams of root *Bb* and *B*. 64
A.3 All chord diagrams of root *C* and *Db*. 65
A.4 All chord diagrams of root *D* and *Eb*. 66
A.5 All chord diagrams of root *E* and *F*. 67
A.6 All chord diagrams of root *Gb* and *G*. 68

List of tables

3.1 The number of songs in top ten genres in the dataset. 14
3.2 Examples of chord representation simplification (in the case of root *C*). . . . 16
3.3 Examples of chord component simplification (in the case of root *C*). 17
3.4 The confusion matrix of uni-chord and bi-chord difficulty ratings. 20

6.1 The song lists in the SP and baseline (BL) condition. 43
6.2 The results of the sight reading task. 45

Chapter 1

Introduction

1.1 Background

Playing musical instruments requires motor skills of fingers and hands. Training for these instruments can be demanding and time-consuming though many people have strong desire to acquire such skills. This problem is more prominent in polyphonic instruments, such as guitars and pianos. These instruments involve playing chords (i.e., harmonic sets of more than two notes). Learners thus have to develop sufficient dexterity to play chords, which can be discouraging.

There are many possible approaches for self-training. Learning materials for self-training include chord books (books illustrating finger placements for chords); music theory books discussing relationships between chords; and textbooks providing comprehensive tutorials and exercises. Many online resources for music training are also available. Examples are chordify[1], GuitarTricks[2] and Yousician[3]. However, the number of available materials is enormous, and learners without experience and knowledge cannot easily identify songs to use for their training. For instance, novice learners may not be aware of which songs are easy or difficult to play. Some songs may have important characteristics for effective training (e.g., songs including chords frequently used in other songs). This motivates me to examine another approach to computer-supported music training services mainly targeting to novice learners.

[1] https://chordify.net/
[2] https://www.guitartricks.com/
[3] https://get.yousician.com/

Musical structures, such as tempo and chord occurrences, are known to be quite different across genres (Pérez-Sancho et al., 2009; Tzanetakis and Cook, 2002) and artists (Ogihara and Li, 2008). I investigated on the Billboard music dataset (Burgoyne et al., 2011) and revealed that chord occurrences follow the power law, confirming their skewed distribution (explained in detail later). I also discovered that most frequently appearing chords are different across genres. This leads me to examine a new training strategy of polyphonic musical instruments (guitars, in this work) by taking a data-driven approach. For example, if a system encourages users to practice chords frequently used in their preferred genres, acquired skills may be easily transferred to other songs in those categories.

I present Strummer, a data-driven approach for chord practice in guitars. Strummer considers characteristics of chords in a song for prioritization within a given music set. These characteristics include: chord difficulty, chord frequency, and chord transition difficulty. It then prioritizes songs including chords and their transitions that are frequently-used and easy-to-play. Learners can go through step-by-step and interactive tutorials that are designed to practice from a chord to the entire song. Tutorial contents vary depending on song sets that learners choose for their practice. Thus, Strummer can offer personalized tutorials for accommodating their preferences.

1.2 Contributions

This work contributes to the field of Human-Computer Interaction and music information processing:

- **Quantitative investigations on the occurrences of chords and their transitions in the Billboard music dataset**: This work examines the characteristics of chord occurrences in 727 songs available in the Billboard dataset. I present quantitative results that show a potential for data-driven approaches to chord practice.
- **A chord and song prioritization algorithm and the Strummer interface**: My algorithm identifies which songs users should practice first based on characteristics of chords. I also demonstrate an interface that provides step-by-step tutorials of songs with my prioritization algorithm. The interface includes audio analysis on chords which users play, enabling interactive feedback on how successfully they can play chords and songs.

- **Algorithm and user evaluations**: An algorithm evaluation confirms that my approach can support skill transfer to a larger set of songs than a method based on the sum of chord difficulty (the baseline method). A user study uncovered that my prioritization algorithm offers similar user experience to the baseline method in terms of perceived difficulty increase.

Chapter 2

Related Work

2.1 Music Practice Support

Playing musical instruments requires a considerable amount of training and effort. Prior work has investigated how computers can support learners' practice of musical instruments. I categorized them into three groups based on their targeted instruments: piano, guitar, and other instruments. I also discuss representative products and services in music game applications.

2.1.1 Piano

In this research field, many systems have been created for pianos. Dannenberg et al.'s Piano Tutor (Dannenberg et al., 1990) is one of the early prominent work in this space. This system computes learners' errors by taking matches with the given score and using training heuristics. It automatically updates learner models based on performance and provides additional training for improving erroneous parts (called remediation in this work). Their goal is to complement traditional piano lessons with their system by using multimodal computer-based techniques. The system has multiple options. Teacher and student can select support styles from the components of the system (e.g., score tracking and visual feedback).

Huang et al.'s Piano touch (Huang et al., 2008) informs learners of finger movements in piano playing through vibrotactile feedback. They intended that learners can practice pianos in daily activities, even if they are not in front of keyboards. Users wear a glove with vibrotactile motors, and the system activates the motors according to the given score. Their study confirmed that the system enabled participants to play songs after a small amount of training.

Recent work employs a variety of displays and visualization techniques. Takegawa et al. developed a system that recognizes learners' fingering and hand positions on the piano (Takegawa et al., 2011). They tackled a usability problem of understanding presented fingering information and overall fingering flows. The system tracks learners' fingers with optical markers and offers visual feedback onto keyboard surface by a projector. The system presents three types of visual contents on the keyboard and screens: detailed execution information, score information, and command information. They confirmed that the system improved the learning effectiveness compared with the lighted keyboard method, which is well-used in this research area.

Prior research has also investigated systems for computer-assisted learning. Yuksel et al.'s BACh focuses on cognitive workload for adaptively changing the level of piano learning tasks (Yuksel et al., 2016). Their system utilizes functional near-infrared spectroscopy (fNIRS) to sense users' cognitive workload level. The system makes learning materials easier or more difficult based on the users' performance played right before. Therefore, this system does not directly check the performance of users and does not point out poor plays. It means that the system does not suggest for users the way to improve performance. Their work reports that the feedback based on cognitive workload improves users' performance of pianos.

2.1.2 Guitar

There are also several projects targeted to guitar practice. Motokawa and Saito created a system to teach finger placements on the guitar strings (Motokawa and Saito, 2006). It captures users' fingers and guitar strings with a Webcam and displays the captured image with superimposing positions to hold for the given chord. However, this work mostly focuses on improving a vision-based technology for tracking a guitar and fingers, and does not include application evaluations. Barthet et al.'s Hotttabs (Barthet et al., 2011) offers online practice materials, containing multimedia contents, including video tutorials and guitar TABs. Their algorithm based on song popularity recommends songs to users. Then, it also recommends guitar TABs picked from guitar TAB search engine. This recommendation is based on the chords extracted from the song. Although this system designed the compatibility between the learners' voluntary exploration of songs and the support to practice songs, this work does not contain any system evaluation.

2.1.3 Other Instruments

Feygin et al. clarified that haptic feedback is useful for motor skill acquisitions (Feygin et al., 2002). This work shed light on the way of haptic feedback as almost all kinds of musical instruments involve body movements and require motor skills. Grindlay et al. investigated the effect of haptic guidance on musical training by conducting experiments on percussion performance (Grindlay, 2008). They showed the usefulness of haptic guidance when people learn the timing of musical notes and velocities. They also concluded that the combination of haptic and audio guidance benefits novice learners especially in the early phase of training.

Johnson et al. created a system to support playing the violin (Johnson et al., 2011). With wearable sensors, this system offers learners how close their arm movements are to those of teachers through vibrotactile feedback. They also designed their second prototype as it gives visual feedback to teachers along with providing haptic feedback to students. They extended their work to wild studies in another paper (van der Linden et al., 2011). They finally concluded that vibrotactile feedback for students encouraged students to evaluate their own performance and recognize their own problems, and also enabled both teachers and students to together discuss body movements.

Ng et al. employed multimodal interfaces to support violin practices and teaching (Ng and Nesi, 2008; Ng et al., 2007). Wearable sensors recognize movements of learners, and the system displays their 3D reconstruction for review. They used auditory feedback for practice support. Their system also executes *sonification*: the use of non-speech audio to convey information. As the auditory time resolution of human is faster than the visual one, they hypothesized that real-time sonification can be appropriate for motor control training. They sonified the angle of the bow and the rhythm of the bowing movements. This sonification can notify learners how well they performed violins in parallel with the visualization. They concluded through preliminary studies that sonification feedback has very positive effects. However, they also reported that some teachers considered it excessively distracts students from the raw sound of violins.

2.1.4 Music Game Applications

Improving the skills of playing musical instruments often requires repetitive practices. Game interfaces can motivate learners to work on such practice. There are commercially available games and systems for encouraging learners' practice, and they greatly inspired some of the Strummer interface design. Guitar Hero[1] uses a simplified guitar device, and users press its buttons as the interface instructs. Rocksmith[2] provides a similar game environment to Guitar Hero, but users can connect a real electric guitar to the system. Yousician is an online training system for pianos, guitars, and ukuleles. It offers step-by-step tutorials and immediate feedback on the user's performance. Yousician is one of the closest systems to Strummer, but I incorporate a data-driven approach for prioritizing chords and songs for guitar practices.

2.1.5 Summary

I discussed representative work in the field of musical practice support. Many researchers have devoted considerable effort to investigate how to support people to play musical instruments. However, there are few systems which put an emphasis on chord playing. Chords have an important role in music: it makes harmony, and harmony makes music. It can make an accompaniment for melodies and enriches the music. Therefore, the practice support for chord playing is highly beneficial for players. Furthermore, although a large part of prior work has demonstrated their new way to guide people to practice musical instruments, they have rarely discussed how the musical contents of tutorials affect the efficiency of learning. For example, the order of learning materials can have an impact on user experience. As I previously mentioned, the amount of materials are plentiful and it is available online. This encourages me to examine a data-driven approach for exploring an uncovered area of this research field. Data-driven approaches can be useful not only guitars but also other polyphonic instruments. I thus focus on the support for chord practice in a guitar with a data-driven approach.

[1] https://www.guitarhero.com/
[2] http://rocksmith.ubi.com/rocksmith/

2.2 Music Performance Support

Prior research has explored interactive approaches to supporting musical performance as well as training. Pachet et al. developed the system that enables a solo performer to make a real-time adaptive ensemble in a piano trio style (Pachet et al., 2013). Their system analyzes what the user is currently playing within no perceivable latency, and generates accompanying music which fits to the types of the performance (e.g., melody, chord, or bass). McPherson et al. focused on a problem that the piano has no ability of pitch control (McPherson et al., 2013). Pitch control is a commonly-used expression technique (e.g., vibrato) in other kinds of musical instruments. The piano or piano-style keyboard is one of the most widely used musical instrument in the world, but it does not have this mechanism. They created a system that enables piano performers to control pitches of every note in an intuitive way. The system measures finger placements on the key surface using capacitive touch sensors. Then, the system alters pitches depend on how much the user moves fingers after pushing a key. Another system is created for reed instrument into which needs players to blow their breath. Kurosawa and Suzuki developed a robot-assisted playing system for a saxophone (Kurosawa and Suzuki, 2010). People can play the saxophone without their fingering motions by using their system. They reported that the system allowed players to perform expressively with acoustic sound and their performance was highly expressive compared to pieces of MIDI music.

These systems described above aim to support more expressive musical performance. Consequently, novice players are not the targeted users. In this thesis, I pursue an approach to support novice players in guitars.

2.3 Pedagogy and Skill Development

2.3.1 Classical Pedagogy and Conventional Textbooks

As previously discussed, various approaches using computers have been taken to support training and expression. However, similar to the number of computational approaches, music teachers have investigated many approaches to improve playing skills and strove to lower the burden for students over many years. Textbooks are the fruits of their efforts, and *pedagogy* is the discipline and practice of teaching that has been made through such attempts. In the area of classical guitars, Glise compiled one of the most prominent pedagogy books (Glise, 1997). In addition to guitar playing techniques, this book describes pedagogical methods, which includes how to arrange a private studio as a free-lance teacher, as an example. The ending part of this book explains musicianship and aesthetics. These ideas are not directly connected with the techniques of how to play musical instruments. However, as pedagogy describes the methodology of *teaching*, it can be beneficial for both teachers and learners.

2.3.2 Dalcroze method

There are some music education methodologies in the world. One of the most widely adopted methodologies is the Dalcroze method (Jaques-Dalcroze, 1930). The Dalcroze method consist of three-fold approaches: eurhythmics, solfége, and improvisation (Abramson, 1980). Eurhythmics is the most fundamental exercise, which teaches the basic musical concepts of rhythms, structure, and emotional meanings through kinematic movements (e.g., tapping a foot). This method is well-established and powerful to enhance the general musical ability, or “the sense of music”, and helps to internalize the concept of rhythms and structures.

2.3.3 Dreyfus’s Skill Development

Skill acquisitions has been an important research area for psychologists. S. Dreyfus and H. Dreyfus proposed a five-stage model of skill acquisition in adults: Novice, Competence, Proficient, Expertise and Master (Dreyfus and Dreyfus, 1980). Later, the five-stage model was revised as Novice, Advanced Beginner, Competent, Proficient and Expert (Dreyfus et al., 1986). They designed these stages with at set of skill criteria: Recollection (context-free or situational), Recognition (experienced or none), Decision (analytic or intuitive), Commitment (detached or involved). These models are useful for designing users in advance of creating a system for skill development.

2.3.4 Summary

It is pedagogically desirable that learners are directly taught by an instructor from the beginning and start with basic principles. However, human instructors are expensive and not always available. Furthermore, supporting self-motivation and independence is important for successful skill acquisition (Percival et al., 2007). The objective of my work is to broaden learner's choices on practice methods, instead of replacing well-established training approaches.

2.4 Music Information Retrieval

Musical information retrieval (MIR) is a research area that focuses on analyzing and extracting musical information from songs. This community has made a substantial effort on building datasets for research activities. Harte et al. developed a context-free chord transition representation (Harte et al., 2005). They also created a dataset using their representation. Burgoyne et al. extended the dataset to include 890 songs, resulting in the Billboard dataset (Burgoyne et al., 2011). Mauch and Dixon examined drum rhythm patterns of 48,176 songs (Mauch and Dixon, 2012). They found that the occurrences of drum patterns demonstrate well-known characteristics in linguistics (e.g., sparsity) as well as unique properties, such as frequent repetition and strong mutual information rates between drum instruments. Pérez-Sancho et al. categorized song genres by examining similarities of chord transitions (Pérez-Sancho et al., 2009). They discovered that chord transitions were a useful feature for genre classification with supervised machine learning methods. Ogihara and Li's study uncovered different tendencies attributed to composers through a similar analysis (Ogihara and Li, 2008).

These projects in MIR motivate me to take a data-driven approach for my application. My work differs from them in focusing on the design of an interactive system to support chord practice in guitars. My system also offers personalized tutorials based on genres and songs which learners choose.

Chapter 3

Chord Dataset

My system uses chord notations of real world songs as practice materials for a guitar. In this chapter, I discuss details and my analyses on the Billboard dataset I use in the Strummer system.

3.1 Dataset Overview

As mentioned in § 2.4, Burgoyne et al. developed the Billboard dataset containing chord notations (Burgoyne et al., 2011). This is one of the largest publicly-available datasets that includes chord notations of popular music. This dataset consists of 890 songs. The songs in the dataset are chosen from *Billboard* magazine's "Hot 100" chart. The Billboard charts are the most popular and traditional music chart in the United States. The Billboard "Hot 100" weekly records top 100 of the most popular music in the country since 4 August 1958. Burgoyne et al. developed an algorithm of selecting the 890 songs in the dataset. They designed the algorithm such that sampled songs would allow researchers to explore general questions about popularity of particular music. This algorithm allows duplication of songs because some songs ranked in the chart over more than a week.

Figure 3.1 shows an example of chord transcription data in the Billboard dataset. Every transcription data of the song has basic information: the title, artist, metre, and tonic of the song. I explain musical terms in § 3.2. As shown in Figure 3.1, the Billboard dataset lacks genre information that I need for later analysis. I used Wikipedia to acquire genre information before parsing. I describe how I obtained the genres of songs in § 3.3. Chord notations appear after the song information in a structured style. As these transcription data

```
# title: You've got a Friend
# artist: Roberta Flack and Donny Hathaway
# metre: 4/4
# tonic: Ab

0.000000000     silence
0.255419501     A, intro, | Ab:maj | Db:maj/5 | Ab:maj | G:hdim7 C:7 |, (synth)
14.013514739    B, verse, | F:min | C:7/5 C:7 | F:min C:7/5 | F:min/b3 C:7/5 F:min . |, (voice
25.853922902    | Bb:min7 | Eb:7 | Ab:maj | Ab:maj |
37.546666666    C, pre-chorus, | G:hdim7 | C:7 | F:min C:7/5 | F:min/b3 C:7/5 F:min . |
49.184761904    | Bb:min7 | C:min7 | Eb:11 | Eb:maj |
60.961020408    D, chorus, | Ab:maj | Ab:7 | Db:maj | Db:maj |
72.560770975    | Ab:maj | Ab:maj | Eb:7 | Eb:7 |
84.028004535    | Ab:maj | Ab:maj7 | Db:maj | Gb:maj Db:maj . . |
95.393854875    | Db:maj7 C:min7 | Bb:min7 | Ab:maj |
103.635941043   E, bridge, | Gb:maj(9) | Db:maj | Ab:maj | Ab:maj |
114.834331065   | Db:maj | Gb:7 | F:min7 | Bb:min7 | Eb:11 | Eb:maj |
```

Figure 3.1 An example of chord notation in the Billboard dataset.

are woven with chord notations and structural information, it is hard to utilize directly. I thus built a parser and analyzer for this chord dataset. I describe the details of my parser and analyzer in § 3.4.

In Strummer, it provides step-by-step tutorials of chord playing using chord data. I analyzed the traits of the dataset for showing the potentials of a data-driven approach. I explain analysis results in § 3.5. Strummer also needs information of chords difficulty to prioritize learning materials. At the end of this chapter, I describe the details of how I estimated difficulties of chords and its transitions in § 3.6.

3.2 Terminology

As this work involves music information analysis, I provide explanations of technical terms in this section. I also depict structural information of the annotation data in Figure 3.2.

- **Chord**: It is a harmonic set of three or more notes played simultaneously. As I mentioned in § 2.1, chord has a harmony, which is one of the three basic elements of music. Particularly in the popular music, a large part of songs has a harmony of chord progressions. Chords have a significant role in polyphonic musical instruments. They are named according to its root note and their harmonic types. Root notes are only twelve classes of pitches, but there are many types of chords. Most fundamental and important chord types are major and minor. Furthermore, the difference of intervals in chords makes various timbres of harmonies and many kinds of names.

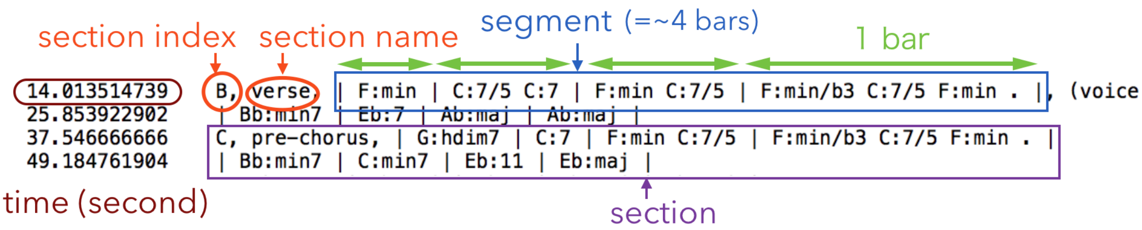

Figure 3.2 An example description of annotation data (song: "*You've Got A Friend* " by Roverta Flack and Donny Hathaway).

- **(Chord) Diagram**: It represents finger placement in a similar fashion to Guitar TAB (the left end of Figure 3.6). In general, people can play a chord in different finger placements in the guitar [1]. Proficient guitar players can use them depending on musical contexts and finger placement of before and after chords. It also enriches music because these chords have its own characteristics of timbre although they are written in same chord symbol. However, it may confuse novice players. I thus assume that a chord has only one hand form leading to a unique diagram. I listed all diagrams used in the Strummer system in the appendices. Each diagram listed in the appendices is one of the representative forms.

- **Metre**: It means the time signature of the song (e.g., 4/4, 6/8). Strummer assumes 4/4 time signature because novice learners tend to get confused when different time signatures appear in practicing chords.

- **Tonic**: It is the central note in the key. For instance, the tonic is *C* if the key of a song is *C major* or *C minor*. In reality, songs in particular keys are easy to perform depending on every instrument. This tonic information would be powerful for guitars because it is easy to transpose keys by using a particular tool (i.e. *capotasto*). However, it also has a disadvantage; the same chord would be played by different finger placement due to handling the tool. In the Strummer system, I assume that learners play chords without using such tools and transposing keys. I thus do not use the tonic information in the current Strummer system.

- **Beat**: It is the most fundamental temporal unit of music. Because beats usually correspond to quarter notes of music and are almost regularly spaced, many temporal features, such as bars and rhythms, depend on beats.

[1]It results in the difference of *open chord*, *low position chord*, and *high position chord*.

Table 3.1 The number of songs in top ten genres in the dataset.

Genre	# of songs	Genre	# of songs
1. Rock	168	6. Pop rock	59
2. Pop	133	7. Hard rock	58
3. R&B	108	8. New wave	39
4. Soul	79	9. Funk	38
5. Country	74	10. Disco	23

- **Bar**: It forms the hierarchical structure of music together with beats. When the time signature is 4/4, each bar consists of four beats (four quarter notes). In the Billboard dataset, an element divided by vertical bars corresponds to one bar and generally contains one chord though there are more in some cases. As a bar containing more than three different chords can be difficult to practice, I limit them up to two of chords and divide them into two half notes.

- **Section**: This represents a portion of a song, such as intro, verse, bridge (or pre-chorus), and chorus. I also use this information for detecting repeated patterns.

- **Segment**: I define a segment as a part of a section that contains four bars. Some songs may have the last segment containing fewer than four bars.

- **Uni-chord and bi-chord**: I define them as a sequence of one and two chords in a song, respectively. To examine quantitative characteristics of chord occurrences, I introduce the notion of n-gram, a well-known feature used in text analysis (Brown et al., 1992), similar to the study by Pérez-Sancho et al. (Pérez-Sancho et al., 2009). Uni-chords represent occurrences of independent chords while bi-chords consider transitions between them.

3.3 Genre Determination

Prior MIR research demonstrated different methods for estimating music genre from various features (Lidy et al., 2007; Nanni et al., 2016). As accurate genre estimation is beyond the scope of this work[2], I decided to take a naïve approach, using descriptions available

[2]Genre classification tasks are usually supposed to prepare a set of genres. I do not presume a set of genres in the genre decision process.

in Wikipedia. I made a genre extractor that automatically searches Wikipedia articles by the title and artist of the given song. It extracts the genre information on the top Wikipedia search result, and tags it to the song. When descriptions in Wikipedia have multiple genres, I associate up to three of them to the song. Table 3.1 shows the number of songs in top ten of genres in the dataset.

3.4 Dataset Parsing and Chord Simplification

As I mentioned in § 3.1, the Billboard dataset allows duplication of songs. Before moving to further data processing, I eliminated 151 duplicated songs from the dataset because completely same materials would not offer additional learning benefits. I also decided to remove twelve songs that only contain fewer than three kinds of chords because they are considered to have little training effects. After the removal of these songs, the dataset contains 727 unique songs. I then investigated the characteristics of chord occurrences in the 727 songs in the Billboard dataset.

In pre-processing, the parser first extracts the basic information of a song, including the title and artist. It then parses the chord information that contains timing and chord symbols. Most songs in the music dataset have repeated *section* patterns (e.g., verses and choruses). The parser performs simple matching of descriptions available in the dataset to mark such patterns.

The dataset follows the chord notation developed by Harte et al. (Harte et al., 2005). Their chord notation syntax is written in the Backus-Naur form; a notation method in context-free grammars. Their syntax adopts parentheses to allow annotators to describe precise components of a chord (e.g., C: (3, 4, 5)). Although this is an accurate notation, it is unnecessary information for playing the guitar. Furthermore, some tension chords (extended from the basic seventh chord) occur rarely in the dataset. These types of chords are also very difficult for novice learners. I thus performed two kinds of simplification before analysis.

3.4.1 Representation Simplification

At first, I performed simplification on chord representations. Table 3.2 shows examples of chord representation simplification. The left column of the table shows all fourteen types of chords I have in the system. In the Billboard dataset, These types of chords are mainly

Table 3.2 Examples of chord representation simplification (in the case of root *C*).

Chord Type	Notation in the dataset	Post-simplification
Major	C:maj	C
Minor	C:min	Cm
Diminished	C:dim	Cdim
Augmented	C:aug	Caug
Seventh	C:7	C7
Major 7th	C:maj7	CM7
Minor 7th	C:min7	Cm7
Minor Major 7th	C:minmaj7	CmM7
Half Diminished 7th	C:hdim7	Cm7b5
Add 9th (Suspended 2nd)	C:add9, C:sus2	Cadd9
Suspended 4th	C:sus4	Csus4
Power Chord	C:5	C5
Major 6th	C:maj6	C6
Minor 6th	C:min6	Cm6

written in the style of the center column of the table. The right column shows the chords after the simplification. In the Strummer system, I employed the styles of chords shown in the right column of the Table 3.2.

3.4.2 Component Simplification

I further executed chord simplification on complex chords: mainly tension chords except add 9th chords. I also converted inversions and fraction chords into simpler forms. Table 3.3 shows examples of how the parser simplifies complex chords. I do not claim this component simplification is truly accurate in the aspect of musical contexts. As I prioritize usefulness for novice guitar learners, I decided to use this component simplification process.

These two simplification processes resulted in fourteen types of guitar chord symbol representations commonly used among guitar players. The right column of Table 3.2 shows the fourteen types of the root *C* chords that I use in the system. After finishing all processing, the parser then converts all information into a JSON file in the format compatible with the Songle Widget (Goto et al., 2015). Figure 3.3 shows an example of the JSON file. I use this converted data in my Strummer system.

Table 3.3 Examples of chord component simplification (in the case of root *C*).

Chord Type	Notation in the dataset	Post-simplification
Root Note	C:1	C
Diminished 7th	C:dim7	Cdim
9th	C:9	C7
Major 9th	C:maj9	Cadd9
Minor 9th	C:min9	Cm7
Major 13th	C:maj13	C
Minor 13th	C:min13	Cm
Chords with brackets (additional note)	C:9(13,#11)	C7
	Csus4(b7)	Csus4
	C:7(b9,11)	C7
	C:min(11,9)	Cm
	C:1(b3,b7,11,9)	C
Inversions and Fraction Chords	C:maj/3	C
	C:min/5	Cm
	C:7/b7	C7
	C:maj/11	C

3.5 Analysis Results

The dataset contains 163 different chords after the simplification, and their total occurrences are 94,618. My analysis discovered skewed distributions of occurrences in both uni-chord and bi-chord. I explain representative observations with songs in pops and hard rocks. I chose pops because it is the second largest genre in the dataset. Hard rock is another genre and is known to be quite different from pop songs.

Figure 3.4 shows the log-log charts of chord ranks and occurrences. A regression analysis confirms that the distributions fit well to the power law. The goodness of fit was .80, .80, and .82 for all genres, pops, and hard rocks, respectively. This result supports the hypothesis that uni-chord occurrences follow the power law, encouraging my exploration on the data-driven approach. Although the chord distributions of both genres are similar, their rankings are quite different. For instance, *F* and *Db*, are among the top ten frequently-used chords for pops, but not for hard rocks. The top ten chords for hard rocks include *B* and *Am*.

```
"chords": [
  {
    "name-triad": "Ab:maj",
    "bar": 0,
    "index": 0,
    "name-full": "Ab:maj",
    "popular-style": "Ab",
    "degree-triad": "I:maj",
    "degree-full": "I:maj"
  },
  {
    "name-triad": "Db:maj",
    "bar": 1,
    "index": 1,
    "name-full": "Db:maj",
    "popular-style": "Db",
    "degree-triad": "IV:maj",
    "degree-full": "IV:maj"
  },
  {
    "name-triad": "Ab:maj",
    "bar": 2,
    "index": 2,
    "name-full": "Ab:maj",
    "popular-style": "Ab",
    "degree-triad": "I:maj",
    "degree-full": "I:maj"
  },
```

```
"repeatSegments": [
  {
    "symbol": "A",
    "repeats": [
      {
        "length": 4,
        "start": 255,
        "id": 0
      }
    ],
    "struct": "intro"
  },
  {
    "symbol": "B",
    "repeats": [
      {
        "length": 11,
        "start": 14013,
        "id": 5
      }
    ],
    "struct": "verse"
  },
  {
    "symbol": "C",
    "repeats": [
      {
        "length": 10,
        "start": 37546,
        "id": 17
      }
    ],
    "struct": "pre-chorus"
  },
```

```
"song": {
  "title": "You've got a Friend",
  "year": 1971,
  "beat": 4,
  "tonic": "Ab",
  "id": 4,
  "metre": "4/4",
  "genre": [
    "folk rock",
    "soft rock"
  ],
  "artist": "Roberta Flack and Donny Hathaway",
  "bpm": 82
}
```


Figure 3.3 An example of the converted JSON file. This example shows the data for the song described in Figure 3.1 and 3.2. The data structure has three components: *chords*, *repeatSegments*, and *song*. The *Chords* part has all chord information of the song and also has chronological information (e.g., “time”, “index”). The *RepeatSegments* part includes structural information. It is grouped by section names. It describes how many times each section repeats in the song. The *Song* part has basic information, such as title and artist.

I can observe a similar trend in bi-chords. Figure 3.5 presents the bi-chord occurrences in pops and hard rocks. Again, the distributions fit the power law well: the goodness of fit of .93 for all genres, pops, and hard rocks.

3.6 Chord and Transition Difficulty Rating

3.6.1 Chord (Uni-chord) Difficulty Rating

As my application is designed for guitar practice, I also need to consider how difficult each chord transition is to play. I first determined chord (uni-chord) difficulty through manual

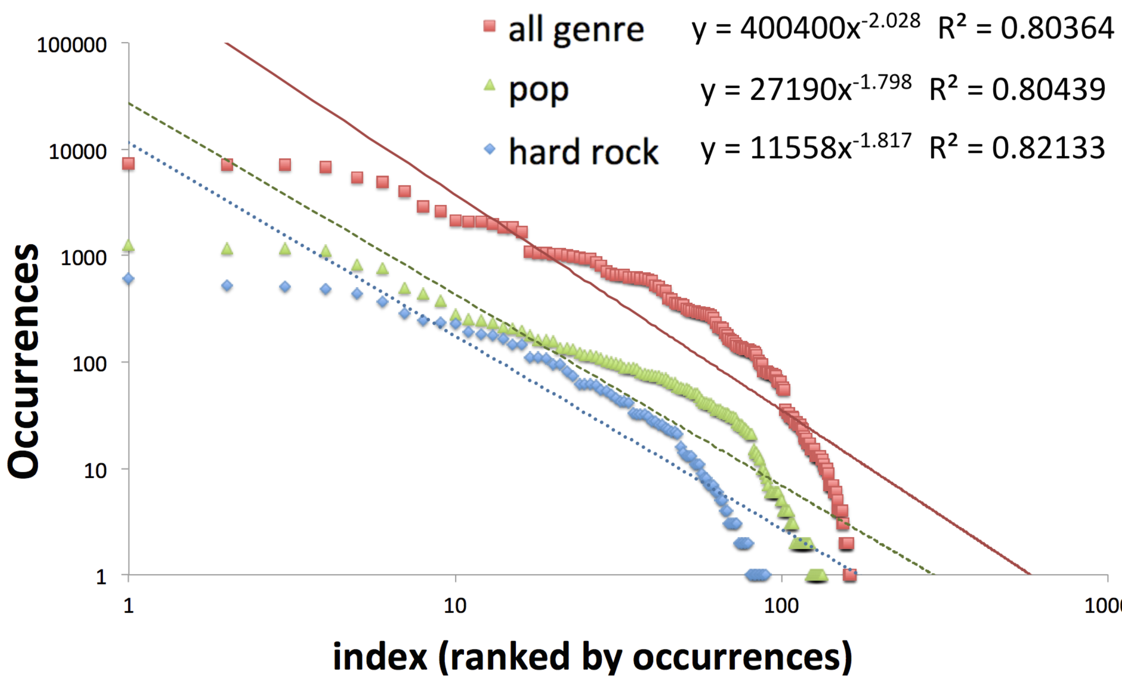

Figure 3.4 Log-log plots of uni-chord occurrences.

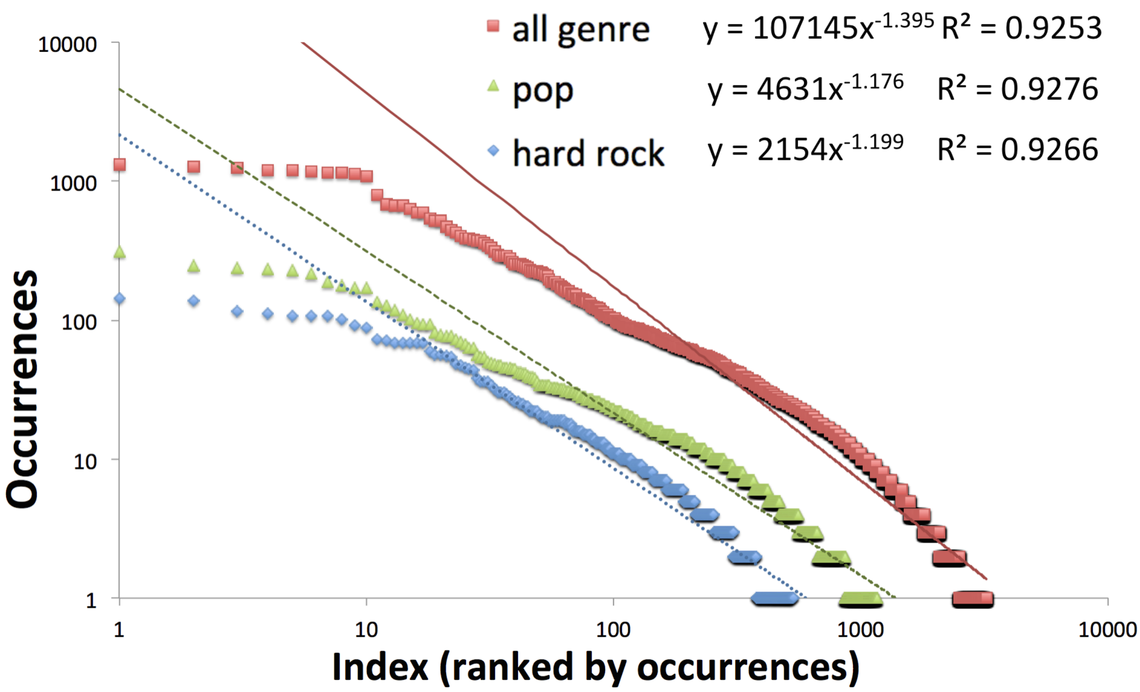

Figure 3.5 Log-log plots of bi-chord occurrences.

Table 3.4 The confusion matrix of uni-chord and bi-chord difficulty ratings.

(a) uni-chord

	rater B					
	Rating	1	2	3	4	5
rater A	1	15	20	1	1	0
	2	0	9	3	1	0
	3	0	6	33	13	0
	4	0	0	17	25	1
	5	0	0	1	22	12

(b) bi-chord

	rater B					
	Rating	1	2	3	4	5
rater A	1	18	11	4	1	0
	2	1	19	8	1	0
	3	0	9	21	8	0
	4	0	0	6	19	1
	5	0	0	0	3	1

rating. Two independent raters, including myself, subjectively evaluated chord difficulty with a 5-Likert scale (1: easiest – 5: most difficult). The raters were instructed to consider only the complexity of the finger placement for each chord. They rated 180 chords including 163 appeared in the dataset. Table 3.4a shows a confusion matrix of two raters. As a result, 52.2% of the chords had the same ratings. I measured how raters' ratings coincide by Cohen's Kappa coefficient. The square-weighted Cohen's Kappa was $\kappa = .81$ (95% CI: [.7534, .8537]), indicating an almost perfect agreement. I use the average rating as a score of chord difficulty in the system. I listed all the chord difficulties resulting from this process in the appendices.

3.6.2 Chord Transition (Bi-chord) Difficulty Rating

I also developed difficulty ratings for chord transitions. However, the number of chord transitions that can be appeared in the dataset is theoretically 26,569, and it is infeasible to rate all of them manually. I thus decided to use a statistical approach to determining chord transition difficulty (bi-chord difficulty). I obtained 5-Likert scale ratings for 100 most frequently-used transitions in a similar manner to the chord difficulty through manual labeling by the same two raters. In addition, I randomly chose 30 of chord transitions and rated in a similar manner to broaden the coverage of my model. Table 3.4b describes a confusion matrix of bi-chord difficulties ratings. Through our manual ratings, 60.0% of the 130 transitions had the same difficulty scores. The square-weighted Cohen's Kappa was $\kappa = .76$ (95% CI: [.6615, .8359]), indicating a substantial agreement.

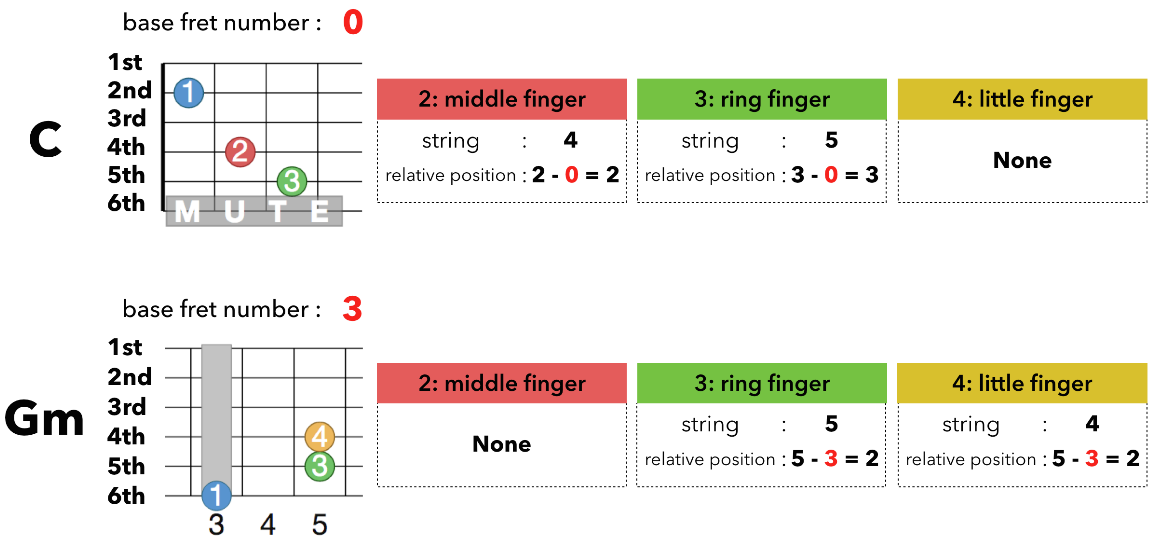

Figure 3.6 A description of finger positions in non-barre and barre chord (*C* to *Gm*).

3.6.3 Chord Transition Difficulty Model

Using these sampled transitions with ratings, I ran a linear regression to create a model which can estimate the difficulty of all chord transitions. I initially started with two potential independent variables and used AIC for determining the optimal model. As a result, I had the seven of independent variables and estimated coefficient values ($p < .001$ for all) besides the intercept of 0.790. I explain how I calculate these variables by taking an example of transition (*C* to *Gm*) shown in Figure 3.6. Note that the value described in the parentheses next to each variables is the estimated coefficient.

1. $\boldsymbol{diff\_t}$ (0.310): The difficulty of the target chord. In this example, the difficulty of *Gm* is 3.0. Thus $diff\_t$ is 3.0.

2. ***fig_num*** (0.211): The increase or decrease of the number of middle, ring, and little fingers (ranged from -2 to 2). I excluded the index finger for this count as it is involved in more than 80% of the chords. In this example: $fig\_num = 2 - 2 = 0$.

3. $\log(\boldsymbol{move\_base} + 1)$ (0.336): The natural logarithm of the absolute movement of left hand, defined as the difference in the base fret numbers. If the origin or target chord is barre chord, I use the number of the index finger fret position as a value. Note that base fret number is zero when it is non-barre chord. In the example of Figure 3.6, the base fret number of *C* and *Gm* is 0 and 3, respectively. Therefore, $move\_base$ is $|3 - 0| = 3$.

4. ***move_figR*** (0.233): The movement of the ring finger. It is defined as the Euclidean distance of the ring finger's position between the origin and target chord. To make this variable independent from *move_base*, I decided to use the relative fret positions from the barre finger. In the example, the target chord *Gm* is a barre chord. Therefore, the relative fret position of the ring finger is $5-3=2$ because the base fret number of *Gm* is 3. Now I can calculate *move_figR* by using the string number and fret number of the target and origin chord: $\sqrt{(5-5)^2+(2-3)^2}=1$. Note that it is zero when the origin or target chord does not involve the ring finger.

5. ***move_figL*** (0.242): The movement of the little finger. The definition is similar to *move_figR*. In this example, *C* chord in this style does not involve the little finger. Thus *move_figL* is 0.

6. ***barre_t*** (0.481): The target chord is a barre, and the origin chord is a non-barre (1 if true, otherwise 0). I added this variable for weighing the transition from non-barre chords to barre chords. In the example, the target *Gm* is a barre chord, and origin *C* is a non-barre chord. Thus, *barre_t* is 1.

7. ***slide*** (-1.401): Both origin and target chords have the same barre form, and furthermore, both of *move_figR* and *move_figL* are 0. (1 if true, otherwise 0). In our case, the origin chord is a non-barre chord; therefore, *slide* is 0.

Eventually, I can get the transition difficulty (TD) of *C* to *Gm* by applying my model:

$$
\begin{aligned}
TD(C \rightarrow Gm) &= 0.790 + 0.310 * diff\_t + 0.211 * fig\_num + 0.336 * log(move\_base + 1) \\
&\quad + 0.233 * move\_figR + 0.242 * move\_figL + 0.481 * barre\_t - 1.401 * slide \\
&= 0.790 + 0.310 * \mathbf{3.0} + 0.211 * \mathbf{0} + 0.336 * log(\mathbf{3} + 1) \\
&\quad + 0.233 * \mathbf{1} + 0.242 * \mathbf{0} + 0.481 * \mathbf{1} - 1.401 * \mathbf{0} \\
&= 2.900
\end{aligned}
$$

The model suggests interesting insights on factors of chord transition difficulties. For example, when the target chord is hard to play, the transition is also likely to be difficult. Similarly, the changes in the number of fingers and their positions also contribute to transition difficulty. If the target chord is a barre, the transition becomes difficult. But if the origin chord also has the same barre, the user may simply slide the hand, resulting in lower difficulty as the negative coefficient value of *slide* implies.

This model achieved high R-squared value, $R^2 = .786$. I use this model to estimate the difficulty of the given chord transition. Note that there are several necessary operations as follows:

- If given chord transition is already rated by the raters, use average rating as TD.
- If estimated TD overs 5.0, TD is 5.0.
- If estimated TD falls below 1.0, TD is 1.0.

Chapter 4

Chord and Song Prioritization Algorithms

The Strummer system takes a data-driven approach for chord practice in guitars. Strummer considers characteristics of chords in a song for prioritization within a given music set. These characteristics include: chord difficulty, chord frequency, and chord transition difficulty. It then prioritizes songs including chords and their transitions that are frequently-used and easy-to-play. In this chapter, I describe why and how I establish these prioritization algorithms.

4.1 Chord Primariness

Strummer needs to identify which chords and songs learners should practice first. This information determines which songs they should prioritize for practice. To this end, I first define desired characteristics of a metric I call chord primariness (*CP*) as follows:

- A chord has high *CP* when it is frequently used and easy to perform.
- A chord has lower *CP* when it is either rarely used or difficult to perform.
- A chord has lowest *CP* when it is both rarely used and difficult to perform.

A high *CP* value means that such a chord plays a primary role in the given song set. In addition, I have the following requirements:

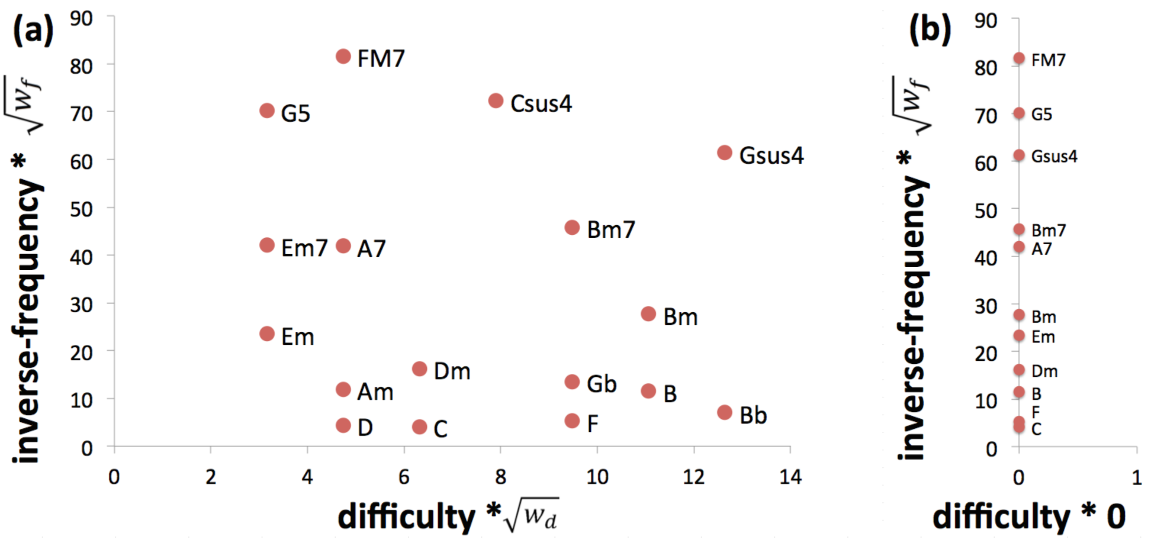

Figure 4.1 The map of representative chords in the *CP* space. (a) The mapping with $w_f = 0.1$ and $w_d = 10$. (b) The mapping with the difficulty ignored.

- The calculation of *CP* must run in real time to maintain interactive responses in the Strummer interface.
- *CP* must be able to adjust weights for the two features depending on applications. For example, some applications may not want to consider chord difficulties (e.g., assuming that learners are already able to play most of the chords). In such cases, the algorithm should calculate *CP* in the same manner besides simply reducing the weight for chord difficulties.
- *CP* must be flexible to include other features, such as similarity of finger placement and frequencies in related genres, for future improvements.

I started metrics commonly used in information retrieval, such as TF (term frequency) and TF-IDF (term frequency - inverse document frequency). However, I immediately found that these metrics are hard to accommodate the needs explained above. For example, TF-IDF can define the uniqueness of chord occurrences but does not have a direct way to consider difficulty.

Instead of developing a metric that directly aggregates features, I take an approach to treating them independently. I map all chords in the coordinate of difficulty and inverse frequency of occurrences. The frequency of occurrences of the chord c_i is calculated as:

$$f_{c_i} = \frac{N_{c_i}}{\sum_C N_{c_i}} \tag{4.1}$$

The denominator represents the total number of occurrences of all chords in the given song set. Figure 4.1a shows some representative chords in this space. In this mapping, each axis also includes its coefficient ($w_f = 0.1$ and $w_d = 10$). I use the same coefficients for the current Strummer system. The algorithm can ignore a particular feature by simply making the associated coefficient zero. Figure 4.1b shows an example of the mapping when I discard the difficulty feature.

In this coordinate, the closer a chord is to the origin, the more primary role it plays in the given song set. Therefore, I can define CP_{c_i} for the chord c_i with an Euclidean distance from the origin in this coordinate:

$$CP_{c_i} = -\sqrt{w_f\left(\frac{1}{f_{c_i}}\right)^2 + w_d {d_{c_i}}^2} \tag{4.2}$$

where d represents the difficulty of a chord, and w_f and w_d are the associated coefficients. I add the negative coefficient of -1 to achieve the three characteristics I desire.

I can extend this notion to measuring a distance between two chords. A long distance means that the next chord that learners will practice is much less important or harder to play. I define the chord primariness distance ($CPd_{c_ic_j}$ where c_i and c_j represent different chords) as follows:

$$CPd_{c_ic_j} = \sqrt{w_f\left(\frac{1}{f_{c_i}} - \frac{1}{f_{c_j}}\right)^2 + w_d(d_{c_i} - d_{c_j})^2} \tag{4.3}$$

4.2 Song Primariness

With CP, I can calculate the primariness (SP) of each song, indicating how likely it includes chords that are important and easy to learn. I calculate SP by simply summing all CP in a song as follows:

$$SP' = \sum_C CP_{c_i} \tag{4.4}$$

However, this SP' does not consider the difficulty of chord transitions. To take into account the chord difficulties, I revise SP' as follows:

$$SP'' = \sum_{C} CP_{c_i} + \sum_{T_j} TD \tag{4.5}$$

This version of SP (i.e., SP'') yet ignores the effect of time intervals between each chord. If learners have a plenty of time to prepare for the next chord, they may successfully change their fingers. But if not, learners will feel that this transition is difficult. The system contains both of whole notes and half notes. Thus, it is desirable to consider the note value of each chord. Finally, I define SP as follows:

$$SP = \sum_{C} CP_{c_i} + \sum_{T_h} TD + \sum_{T_f} TD \tag{4.6}$$

where C, T_h, and T_f represent all uni-chords, bi-chords in the half note, and bi-chords in the whole note. I separate bi-chords in the half and whole note because the former requires faster finger movements, imposing additional difficulty. For T_f, I also excluded transitions whose difficulty was under the average value (= 2.04). In this manner, I can only penalize highly difficult transitions in the whole note. The maximum computational complexity of SP is $O(n(n+m))$ where n is the average number of chords (10.2) in the given songs, and m is its total number (727). It is thus possible to run this calculation for all songs in real time.

SP may use a similar metric to CP for chord transitions instead of pure difficulty. However, bi-chord frequencies are dependent on those for uni-chords. To avoid over-weighing the chord occurrence frequency, I decided to use the formula 4.6 for calculating SP.

4.3 Song Primariness Quantitative Evaluation

To validate benefits of SP, I define an unseen song to be "$k\%$-playable" if learners would be able to perform $k\%$ of the chords in that song. 100%-playable means that learners have already practiced all chords in the given song. I regard song prioritization as effective if it achieves a higher number of $k\%$-playable with practice in a smaller number of song.

I compared SP against the total chord difficulty (TCD) as a baseline approach (BL). TCD is defined as the sum of difficulty scores of all chords in a song. Sorting in the ascending order of TCD means that the system prioritizes songs that have fewer and easy-to-play chords.

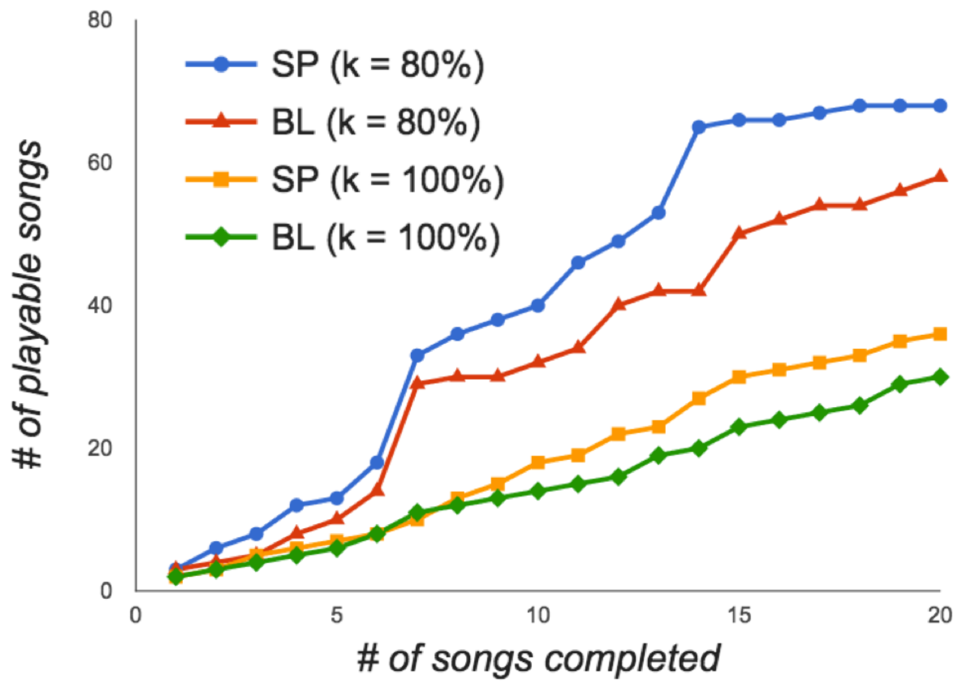

Figure 4.2 Plots of k%-playable songs (genre: pops) across the number of songs completed ($k = 80$ and 100) in two ranking algorithms (*SP* and *BL*).

Figure 4.2 shows the number of 80% and 100%-playable songs under an assumption that learners have completed practicing the first n songs in the two rankings (n: 1 – 20). In both 80% and 100%-playable, *SP* constantly exhibits a higher number than BL. This means that the chords learners would practice can be more useful to play unseen songs in *SP*. Strummer thus uses *SP* to rank songs in the given set and sorts them in a descending order of *SP*.

4.4 Shortest Chord Path

Strummer also needs to determine which chords learners should practice first in the song they are training. I again utilize the chord coordinate shown in Figure 4.1a for this purpose. As explained above, a chord with high *CP* is placed near the origin. Thus, I can order the chords by identifying a path from the origin that traverses all of them.

I define the shortest chord path (*SCP*) as the shortest route that starts from the origin and visits all nodes in the coordinate. A general problem of obtaining shortest paths between two nodes is called Shortest Hamilton Path, and it is known to be NP-complete. However, a problem of finding the *SCP* is considered as its special case, and I can take a polynomial-time solution.

The space I am discussing has three characteristics: 1) the nodes are fully connected with non-negative weights; 2) the start is fixed (i.e., the origin), and 3) the *SCP* needs to visit all of them only once. Without the third requirement, I can use Dijkstra's or Bellman-Ford algorithm (Dijkstra, 1959). But neither of them guarantees that a resulted shortest path travels all nodes.

A further investigation leads me to the conclusion that a greedy algorithm satisfies all requirements. Dijkstra's algorithm relies on the property that any subpath of the shortest path is also shortest. As this holds in the case as well, the algorithm simply needs to traverse a node that is connected at the shortest distance. Algorithm 1 describes a pseudo code to find the *SCP*.

Algorithm 1 Find SCP

```
path ← [null array];
n ← [origin];
nodes ← [all chord nodes];
while nodes ≠ null do
    Calculate CPd_nv for each node v in nodes;
    n ← the node with the smallest CPd_nv;
    Push(nodes, n);
end while
SCP ← path;
```

The computational complexity of calculating the *SCP* with this greedy algorithm is $O(n^2)$ where n is the average number of chords in each song. This is executable in real time.

Chapter 5

The Strummer System

Strummer is an interactive system for guitar chord practice. In this chapter, I explain the details of every tutorial that Strummer provides and the details of implementations.

5.1 Strummer Tutorial Interface

The Strummer interface provides step-by-step tutorials for efficient chord practices. When learners log into the system, they first can choose genres and songs they want to practice. Strummer then displays songs in the chosen set in the descending order of *SP* (Figure 5.1a). This list also provides information about the number of chords and sections in each song. Learners can select a song to start tutorials.

Tutorials in the Strummer system are designed to provide step-by-step practice for the chosen song. As previously mentioned, songs generally have some repetitions (i.e., sections). The system first removes repetitive sections and generates lessons each of which contains a unique set of chords. When learners choose a song, the system presents a set of tutorials, each of which focuses on practicing one of the sections (Figure 5.1b). During a tutorial, Strummer provides six practice stages explained later in this section (Figure 5.1c).

5.1.1 Listen Stage

At the beginning of tutorial, learners are encouraged to listen to the entire section before they start to strum chords. They can listen to the song along with chord information (Figure 5.1d).

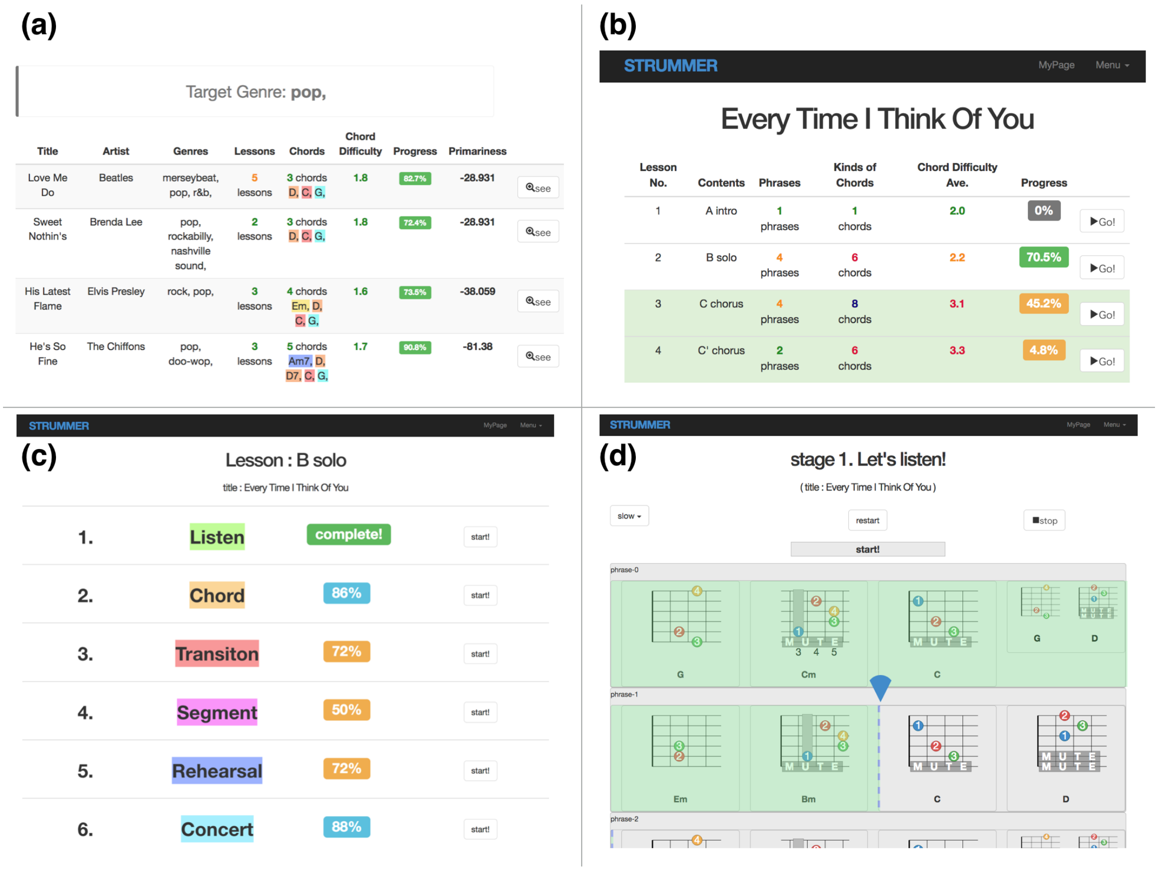

Figure 5.1 Screenshots of the Strummer system interface. (a) A list of songs in the system. It is grouped by pop genre and sorted by *SP*. (b) Each song is divided into sections to make practice manageable. Each section has a *lesson* respectively. (c) Lesson top menu. It has six practice stages. (d) Listen stage. Learners can listen to chords along with its finger placements in the Listen stage.

After learners get familiar with the overall chord structure of the section, they can move to the next stage.

5.1.2 Chord Practice Stage

In this mode, learners practice all chords used in the chosen section one-by-one. Figure 5.2 explains how the chord practice stage proceed. I describe the interaction with an example case where a learner is practicing a *Bm* chord in Figure 5.2. The system sorts the chords by *CP* in the descending order (Figure 5.2a). Therefore, learners start with practicing chords that are frequently used and easy to perform.

When they click a “practice” button corresponding to each chord, the Strummer system displays the finger placement in a similar fashion to Guitar TAB (Figure 5.2b). As I defined

Figure 5.2 An overview of chord practice stage workflow.

in Section § 3.2, I call this finger placement as a *diagram*. The system plays the chord in a strumming style if the "listen" button is clicked. Learners can then confirm how the chord sounds like.

When learners feel ready, they press the "click to start" button to check their performance. After four beats of countdowns, the system then activates an audio-based performance assessment module. I explain its detail in § 5.3. The Strummer interface then informs them whether they successfully played the given chord. Upon unsuccessful performance, it offers feedback about which strings learners failed to play (Figure 5.2c). When they play the chord correctly, the gray box at the lower left of the screen is filled with the word of "OK!" (Figure 5.2e). This "OK!" mark disappears when they play the chord unsuccessfully. If three of the gray boxes are filled with "OK!" mark, the system records the completion (100%).

Providing Additional Lower-level Training

When learners miss the chord more than five times or they miss it three times in sequence, the system enables a button for additional tutorials (yellow button at the upper side in Figure 5.2c). Learners can continue to practice the chord separately in those tutorials (Figure 5.2d to h). These separate tutorials offer lower-level training. It starts with practice for each finger and progressively increases the number of fingers to be used. If the target chord is barre chord, the system increases the number of fingers except for the index finger which forms barre. Figure 5.2d shows the initial scene of this training. The diagram emphasizes the gray box of barre finger and the number 2 in a red circle, which indicates the middle finger of the left hand. The system makes other numbers transparent and makes the first and second strings blinking. When they click the "Click to Start" button, the system only checks the notes of the first and second strings (i.e., Gb4 and D4 in this example).

Similarly to normal chord practice stage, when learners play the displayed partial chord correctly and filled three of gray boxes with "OK!", the system increases the number of fingers and proceeds to next practice (Figure 5.2e). The system empties all the "OK!" boxes, then displays the number 4 in a yellow circle, and makes the first to the third strings blinking (Figure 5.2f). In a similar way, the system progressively increases the number of fingers to push down (Figure 5.2g).

However, learners still may struggle to play a chord even after this additional training. The system records which finger combinations they have succeeded. In the example of Figure 5.2, the learner was not able to play strings from the first to fourth correctly (Figure 5.2h). Then, the system provides the finish button (at the upper side in Figure 5.2h). when she presses the finish button, the system records how many strings she can play as an accomplishment rate (described in percentages). In this case, she was able to play strings from the first to third. Therefore, the system records her progress of this chord as 50.0% (Figure 5.2i).

Complementing Unplayable Notes

When learners can play chords partially, the system applies performance assessment only for the placed fingers in later tutorials. Then, the system makes the sound of these omitted notes on behalf of them. For example, in Figure 5.4a, the diagram of *Bm* is abbreviated and different from the original form. The system provides this feature so that learners can continue their practice even if they are not confident in playing some of the chords.

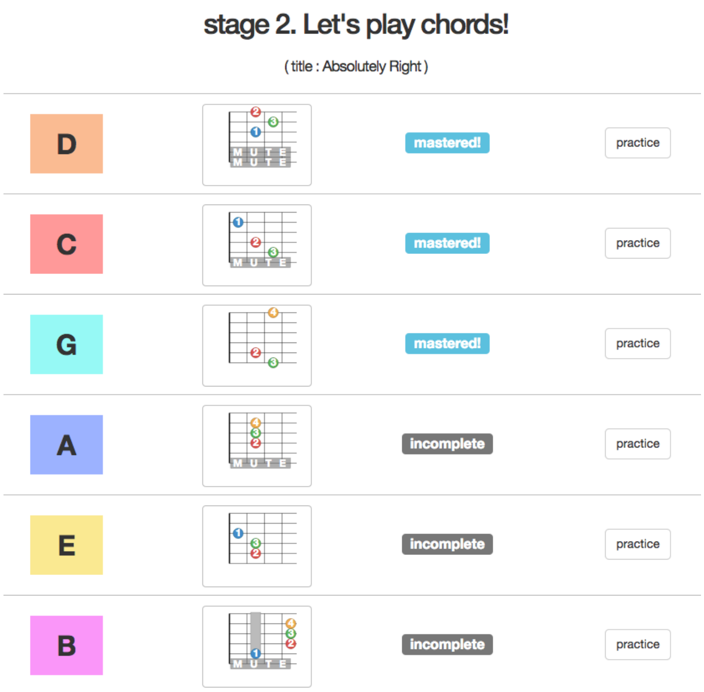

Figure 5.3 An example of chord stage top menu which include mastered chords.

Skipping Mastered Chords

As mentioned above, the system records learners performance after completing the stage. Strummer system is designed for beginners to practice guitar chords starting from easy-to-learn chords. However, it can be annoying if the system requires learners to practice chords that learners already have mastered. Therefore, the system checks how well they can play each chord. If the learners completed the practice on a given chord and achieved 100% progress three times in recent five trials, the system judges they "have mastered" the chord. The system then puts the "mastered" label (Figure 5.3). In this way, they can skip practice of mastered chords.

5.1.3 Chord Transition Practice Stage

Learners can practice chord transitions in a similar manner. Figure 5.4 shows screenshots of this stage. All chord transitions in the chosen section are listed (Figure 5.4a). As explained in § 5.1.2, chord diagrams which they could not achieve 100% progress in the previous chord stage are changed into easier forms. The chord transition practice stage offers the same tutorial as the previous stage except that learners play a pair of chords instead of individual

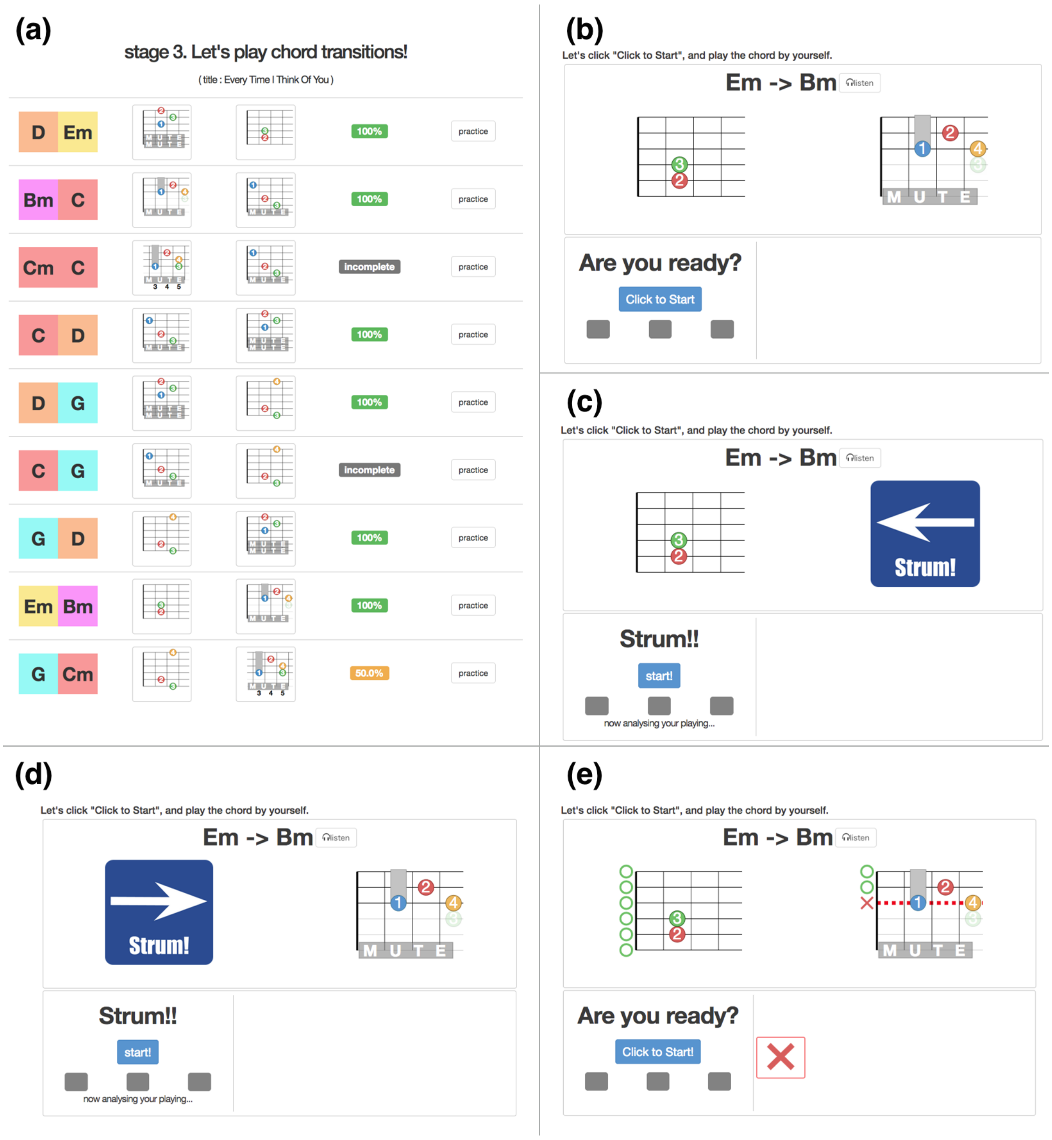

Figure 5.4 An overview of chord transition practice stage workflow.

chord. Figure 5.4b is the initial scene of this stage. When they click the "click to start" button, the system provides two of arrow-shaped indicators along with beats (Figure 5.4c, d). Figure 5.4e means that learners only failed to play the third string of the latter *Bm* chord. The system records the completion (100%) if they played the chord transition successfully and satisfied the same condition in the chord practice stage. Otherwise, the system records a progress of the transition as 50.0%. The system does not offer performance assessment on the latter of this chord transition in later tutorials. Learners who cannot nicely play particular chords might be easily discouraged when they proceed to later stages and face such chords many times. Therefore I decided to offer this supportive function. In Figure 5.4a, the progress

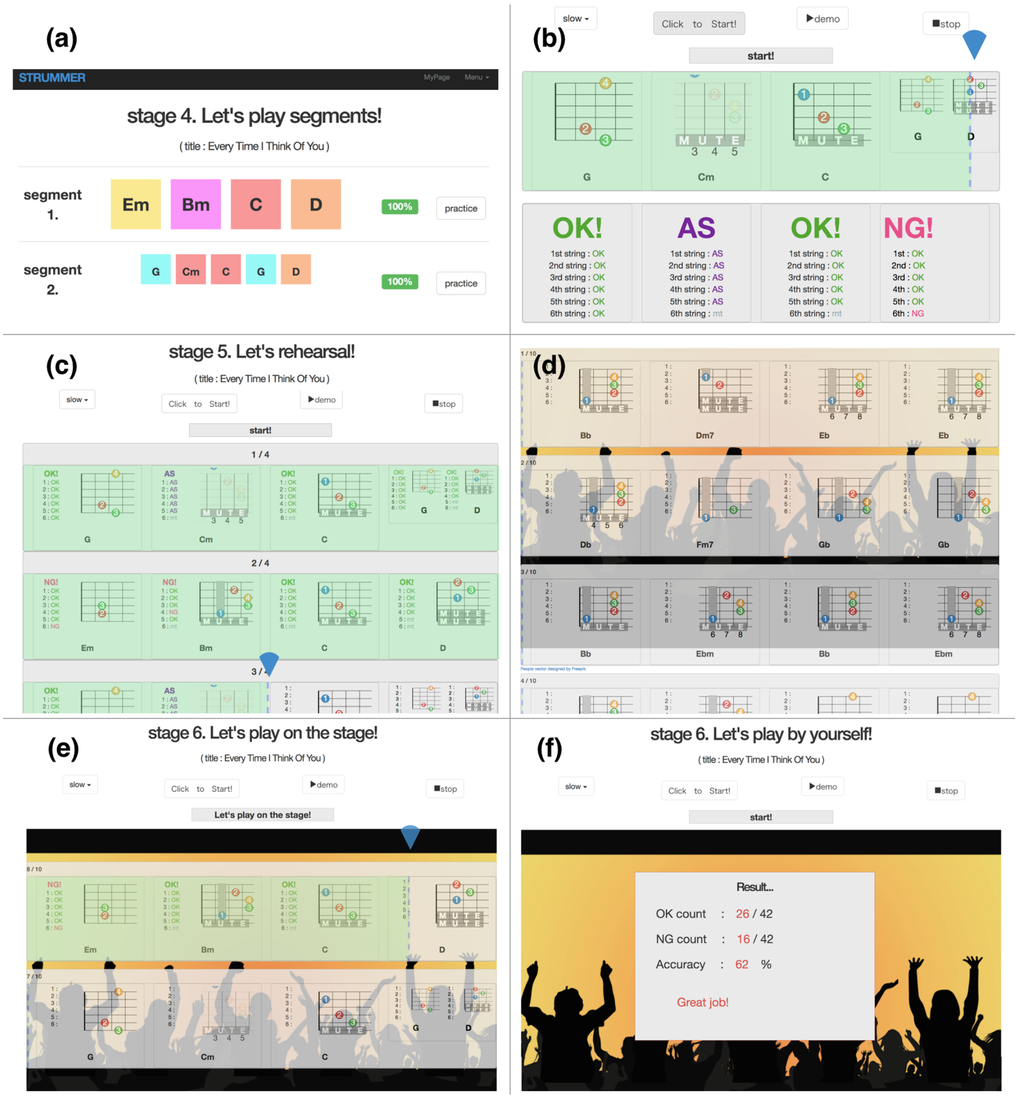

Figure 5.5 Screenshots of the Strummer system interface. (a) A list of segments in the chosen section. (b) Segment practice stage. (c) Rehearsal stage. (d, e, f) Live concert stage.

of the chord transition from *G* to *Cm* is labeled 50.0%. Accordingly, when they come across this *G* to *Cm* transition, the system disables chord analysis of latter *Cm* chord.

5.1.4 Segment Practice Stage

After completing the chord transition practice stage, learners move to practice segments in the chosen section. As I declared in § 3.2, a *segment* is a part of a section. It contains four bars in most cases. Figure 5.5a shows a list of segments in the section. The section that I am explaining as an example contains four segments. However, this list has only two segments because the system has eliminated duplicated segments. Similarly to the previous stage, when they enter to each tutorial, the system displays the chord name and the diagrams (Figure 5.5b). When they press the “click to start” button, an animation starts over the chord sequence, indicating the timing to play. The system also activates the performance assessment module and provides feedback about whether they have successfully played each chord at the right timing. It displays the result of their performance below the diagrams. “AS” is the abbreviation of “assisted” and it means that the complementing function is being activated. As previously mentioned, the system disables the assessment module and synthesizes the sound on behalf of them as scaffolding for the chords or its transitions they have not completed in former stages. In Figure 5.5b, the diagram of *Cm* becomes transparent and the system shows “AS” result. The system does not judge the performance of *Cm* because the user marked the progress of the *G* to *Cm* transition as 50.0% in the chord transition practice stage. In this manner, learners can continue their practice without being constrained to chords they cannot play well. They can complete this stage by correctly playing the segment at least once.

Speed Control

In this stage and later stages, the system provides speed control. Learners can change the speed of the beat indicator by clicking the button at the upper left of Figure 5.5b and choosing from five pre-determined speeds. Speed control affords them to practice at an appropriate speed rate. For example, when they feel that the chosen segment is hard to play at the default speed rate, they can slow down.

5.1.5 Rehearsal Stage

Learners proceed to the rehearsal stage (Figure 5.5c) after they complete practicing segments. In the rehearsal stage, they play the whole chord sequences of the chosen section. This stage is intended to provide additional repetitions in practice, helping skill fixation (Fitts and Posner, 1967). The system marks the stage as completed when they can play all chords successfully. But they can move to the next stage even if they have not completed.

5.1.6 Live Concert Stage

At the end of the tutorial, learners are invited to play the song they have been practicing as if they would do in a concert. I take a similar user interface to popular music games and training systems, such as Guitar Hero and Yousician. In this stage, they are asked to play all sections they have completed as well as what they are currently practicing (Figure 5.5d). In the rehearsal stage, there are four segments to play. However, this concert stage has ten segments to play because of containing other sections that they have already completed in the chosen song. Similarly to segment stage and rehearsal stage, they strum displayed chords along with the indicator (Figure 5.5e). When the indicator comes to the end of each segment, the segment wipes out, and next segment comes up. When they play the indicated chord correctly at the right timing, black silhouetted audience in the background jumps up and gives a cheer as audio feedback. After the performance, the system displays the score based on the number of successfully-played chords (Figure 5.5f). Learners can complete the lesson by being able to perform 60% of the chords.

5.2 Implementation

I implemented the Strummer system in Ruby on Rails and JavaScript. This is because I can easily change contents even after public release if it is implemented as a web application. This feature benefits the Strummer system. The present Strummer interface was tested and implemented on Firefox ver. 50.1.0., but can be immediately available to other common Web browsers.

To accomplish lower-level training and chord complement functions, I need to make the diagrams programmable. I decided to dynamically make diagrams instead of using static image files of all chord finger placements. Therefore, I used SVG (Scalable Vector Graphics) to depict diagrams.

5.3 Performance Assessment Module

For providing feedback of learners' performances, I require that the system can distinguish strings from where the sounds come. It is not sufficient to simply recognize the chord of recorded guitar performance because I want to provide learners with detailed feedback on their performance. Suppose that the chord recognition result is Am even though they intended to play C chord. The components of the Am chord are A, C, E, and the C chord are C, E, G. Therefore, the system can infer that they failed to play the sound of G. However, all strings can generate the sound of G. I need octave information in addition to pitch class information, such as scientific pitch notations [1]. The learners are still left behind about on which string their finger placement need improvements. Therefore, I have to design the module such that it can inform learners of what string is not correctly played. To realize that, the system needs to execute multipitch analysis.

Multipitch analysis is an approach to musical audio processing. However, even recent work cannot fully separate multiple sounds into original notes (Ojima et al., 2016). Furthermore, I attempt to make the system without any additional equipment (e.g., microphones) and run the module in real time. To take into these constraints, I decided to use a naïve approach. After several experimental trials, I consequently adopted the approach of matching the fundamental frequencies of a chord with the observed frequency peaks.

I take an example of the C chord to explain how the module judges what strings are wrong. The components of C chord are theoretically C, E, G notes. As previously stated in Section § 3.2, I assume that one chord uniquely corresponds to one finger placement (i.e., a diagram) in the Strummer system. Furthermore, I can derive pitch information of chord components in scientific pitch notation from the diagrams. Therefore, I can obtain the component notes of every guitar chord used in the system. The C chord in Strummer system consists of the $C3, E3, G3, C4, E4$ notes. Each note is generated from the guitar strings; thus the system can distinguish what strings are played correctly by checking what component notes of the target chord are peaked in frequency.

Fourier Transform is the main tool for my audio signal processing. The system has to do Fast Fourier Transform (FFT) to achieve required functions. FFT in Web API that developers can use in standard web browsers does not take sufficiently long samples for required audio analysis. I thus run the performance assessment module locally on the learners' machine. It is written in Python with the numpy and pyaudio libraries. The sampling rate is 44.1 kHz, and the window size of FFT is 8192 samples. I used the Hanning window in sampling. The module sends the assessment results to the main system via HTTP.

Figure 5.6 shows an overview of the performance assessment workflow. In this figure, suppose that the learners intended to play C chord, but the learners forgot to push the fourth string by her middle finger of the left hand (shown as number 2 in a red circle). The correct placement of middle finger is second fret of the fourth string. In that way, she unintentionally made the sound of $D3$ instead of $E3$. The module recognizes the difference between the

[1]Scientific Pitch Notation is a representation of musical note names and octave numbers simultaneously (e.g., $C3, C4, Db4$). There are several formats, but I use the lowest note used in regular tuning in normal classic guitars, such as $E2$, in this thesis.

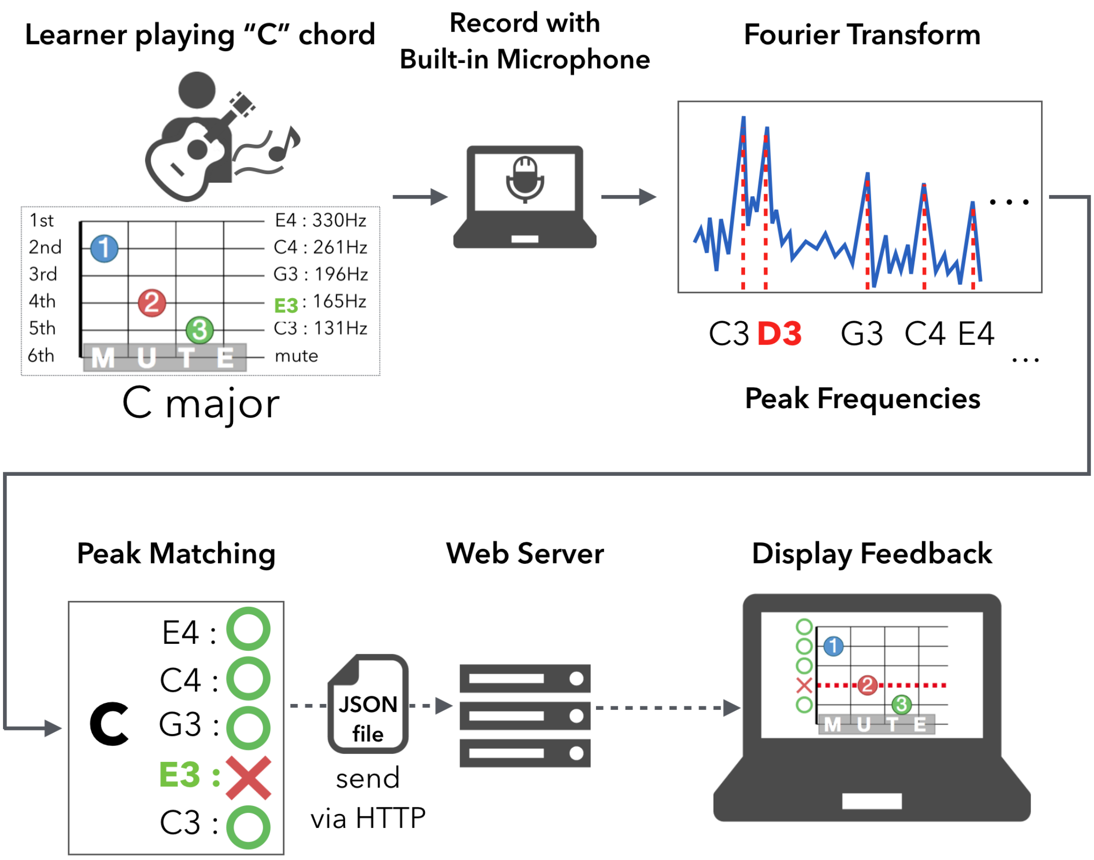

Figure 5.6 An overview of the performance assessment workflow.

components of the correct chord and actual performance, and then sends back the result of to the learners' PC. The system then displays the results in the Stummer interface as feedback.

Chapter 6

User Evaluation

6.1 Objectives of the User Study

In Section § 4.3, a quantitative examination already confirmed an advantage of the algorithms used in Strummer to support efficient skill transfer. But the user experience of Strummer still remains unknown. If the song list sorted by the SP algorithm does not provide smooth progression from easy to difficult songs, learners may still get easily discouraged. I thus conducted a comparative user study to examine how similar the lesson progression with SP would be in comparison to the purely difficulty-based approach.

My main hypothesis in this user study was that perceived lesson progression with the SP algorithm would be similar to the baseline method (using TCD: the total sum of the chord difficulties). In addition, I also investigated the user experience of the features and tutorials to validate the design of the Strummer interface.

The user study involved musical instrument training. A long-term study would be ideal, but it would make comparisons difficult because conditions and contexts for each participant can be very different. Moreover, conventional statistical methods (e.g., hypothesis testing) would not be appropriate to confirm the similarity between the two conditions due to the limited number of participants I would be able to recruit. Therefore, I decided to run a controlled study and mainly take a qualitative evaluation approach.

6.2 Participants

I recruited ten people (all male, the average age of 22.2). One of them had an experience of playing the electric guitar for roughly one week. The rest had nearly no experience of playing the guitar.

I divided the participants into two groups (SP1 – SP5 and BL1 – BL5). The SP and BL groups practiced the songs sorted by the *SP* and the baseline algorithms (*TCD*) respectively through the Strummer system. The participants were not informed which algorithm was used. The rest of the system, including the interface and functionality, was the same between the two conditions.

6.3 Practiced Songs

I chose pops for this user study as it was the second largest genre containing 115 songs and included various types of songs. Table 6.1a and 6.1b show the songs the SP and BL participants practiced during the experiment, respectively. Although there are some similarities between the two groups (e.g., first and second songs), each algorithm expresses its characteristics well. The fourth song in the BL condition (“This house") includes a barre code of *Gbm*. All chords in this song are half notes, making transitions difficult. Because of these factors, this song did not appear in the SP condition.

6.4 Procedure

I designed the experiment to be five days long. I asked participants to come to the user study location every day during the experiment. They were instructed to practice with the given condition for at least one hour. Six of participants were not able to accommodate this schedule, and they practiced for two hours on the fourth day. I lent participants a classical guitar during the experiment.

At the beginning of the first day, I explained the interface and experimental procedure. I also manually provided basic tutorials about how to play the guitar. After this pre-experimental briefing, the participants were asked to practice starting from the first song in the given list.

Table 6.1 The song lists in the SP and baseline (BL) condition.

(a) The songs used in the experiment in the SP condition.

Rank	Title	Artist	Chords	*SP*
1	Love me do	The Beatles	C,D,G	-28.9
1	Sweet nothin's	Brenda Lee	C,D,G	-28.9
3	His latest flame	Elvis Presley	Em,D,C,G	-38.1
4	He's So Fine	The Chiffons	Am7, D, D7, C, G	-81.4
5	No Charge	Melba Montgomery	D,D7,C,G	-96.2
6	Seasons Of The Heart	John Denver	G7, C, G, C7, F	-103.0
7	I Want To Walk You Home	Fats Domino	Ab7, Gb, Db	-125.5
8	I Found A True Love	Wilson Pickett	Am, C, G, C7, F, F7	-129.6

(b) The songs used in the experiment in the baseline (BL) condition.

Rank	Title	Artist	Chords	*TCD*
1	Sweet nothin's	Brenda Lee	C,D,G	5.5
1	Love me do	The Beatles	C,D,G	5.5
3	Till The End Of The Day	The Kinks	A5, Bb5, C5, D5, F5, G5	6.0
4	This house	Tracie Spencer	D,E,Gbm	6.5
4	His Latest Flame	Elvis Presley	Em,D,C,G	6.5
6	No Charge	Melba Montgomery	D,D7,C,G	7.0
7	Fever	Rita Coolidge	Am, Am7, E7, F7	7.5

When participants completed practicing one song, I conducted a short interview with a questionnaire to understand their experience. In addition, I asked Likert scale questions about how difficult the current song was as follows:

- **A**: How difficult was the song compared to the previous song? (1: Much easier – 7: Much more difficult)
- **B**: How tedious were the tutorials compared to the previous song? (1: Not at all – 5: Extremely tedious)
- **C**: How difficult was the song compared to the most difficult song so far? (1: Much easier – 7: Much more difficult)

Figure 6.1 A scene of sight reading task in the user study.

- **D**: How tedious were the tutorials compared to the most difficult tutorials so far? (1: Not at all – 5: Extremely tedious)
- **E**: Was there any frustrating or discouraging moment in practicing this song? (1: Not at all – 5: Almost every moment)

Furthermore, I conducted another short interview to elicit any opinion about the system at the end of each day.

On the last day, I asked participants to perform a new song in the same genre that they had not practiced (Figure 6.1). The intention of this sight reading task was to emulate the experience of playing at sight, requiring sufficient skill fixation on chords. I also wanted to ensure that participants learned how to play chords. I chose "*(SittingÓn) The Dock Of The Bay*" by Otis Redding. This song requires quick transitions in frequently-used chords (C, D, G), and also includes bar chords (F, B) and unseen chords (A, B). After 15-minute practice, they played the song in front of me. I audio-taped their performance for later analysis.

Interviews were not in English because all the participants were neither native nor proficient in the language. I transcribed and translated the interviews to English as faithfully as possible for the report in this paper. I offered participants approximately 50 USD in local currency as compensation at the end of the study.

Table 6.2 The results of the sight reading task.

Chord	Em	D	A	C	E	G	F	B	Total
Occurrences	2	4	3	5	1	14	1	2	32
Difficulty	1	1.5	2	2	2	2	3	3.5	2.13(ave.)
SP1	2	4	3	3	1	14	0	1	28
SP2	1	4	0	3	0	12	1	0	21
SP3	1	0	0	3	0	12	0	0	16
SP4	2	4	3	5	1	14	0	0	29
SP5	2	4	3	5	1	13	1	2	31
BL1	2	4	3	5	1	14	1	1	31
BL2	2	4	3	5	0	14	0	0	28
BL3	2	2	0	3	1	11	0	0	19
BL4	2	3	0	1	0	12	0	0	18
BL5	1	4	2	5	0	14	1	1	28
Success Rate (%)	85.0	82.5	56.7	76.0	50.0	92.9	40.0	25.0	77.8

6.5 Evaluation Results

6.5.1 Sight Reading Task Performance

All participants successfully completed the study. I evaluated the performance of the sight reading task. This evaluation mainly focused on how well the participants were able to hold down chords by their left hands at the right timing. Therefore, I decided that the performance of each chord is acceptable when the participants held down the strings correctly and produced right sounds even if the sounds were not very clear. Table 6.2 presents the number of successful chord playing in sight reading tasks. All participants performed at least half of all chords well. Participants were able to play even new chords at a reasonable accuracy given the short amount of practice (15 minutes). To investigate the user experience induced by the SP and baseline algorithms, I examined qualitative evidence obtained from the participants.

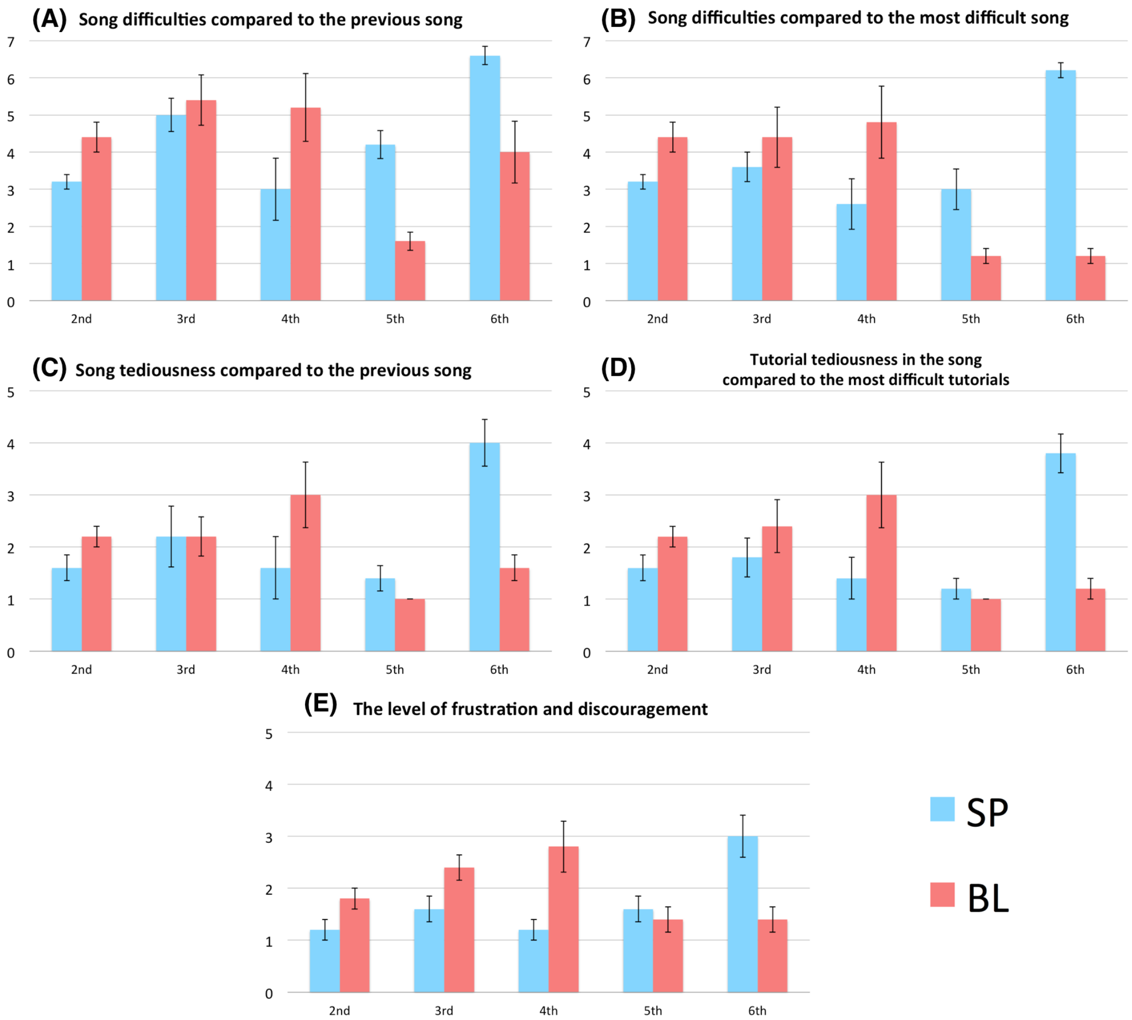

Figure 6.2 Questionnaire results. Each figure labeled with capital letter corresponds to each question and it represents average scores of the SP and BL participants. The horizontal axis indicates the second to the sixth song they practiced in both conditions. The error bars represent standard error.

6.5.2 Questionnaire

The primary objective of this user evaluation was to examine perceived similarity between the SP and baseline algorithm. I looked into how often the participants felt a large difficulty change during the experiment. Figure 6.2 shows the overall results. Figure 6.3 presents the breakdowns of the results.

I further investigated the perceived smoothness in lesson progression in the two methods. I divided the 7-Likert scale responses of the question A into two groups. The responses with the score of 3 – 5 and the rest were regarded as smooth and hard perceived progression, respectively. All SP and BL participants completed the sixth song. For a fair comparison, I

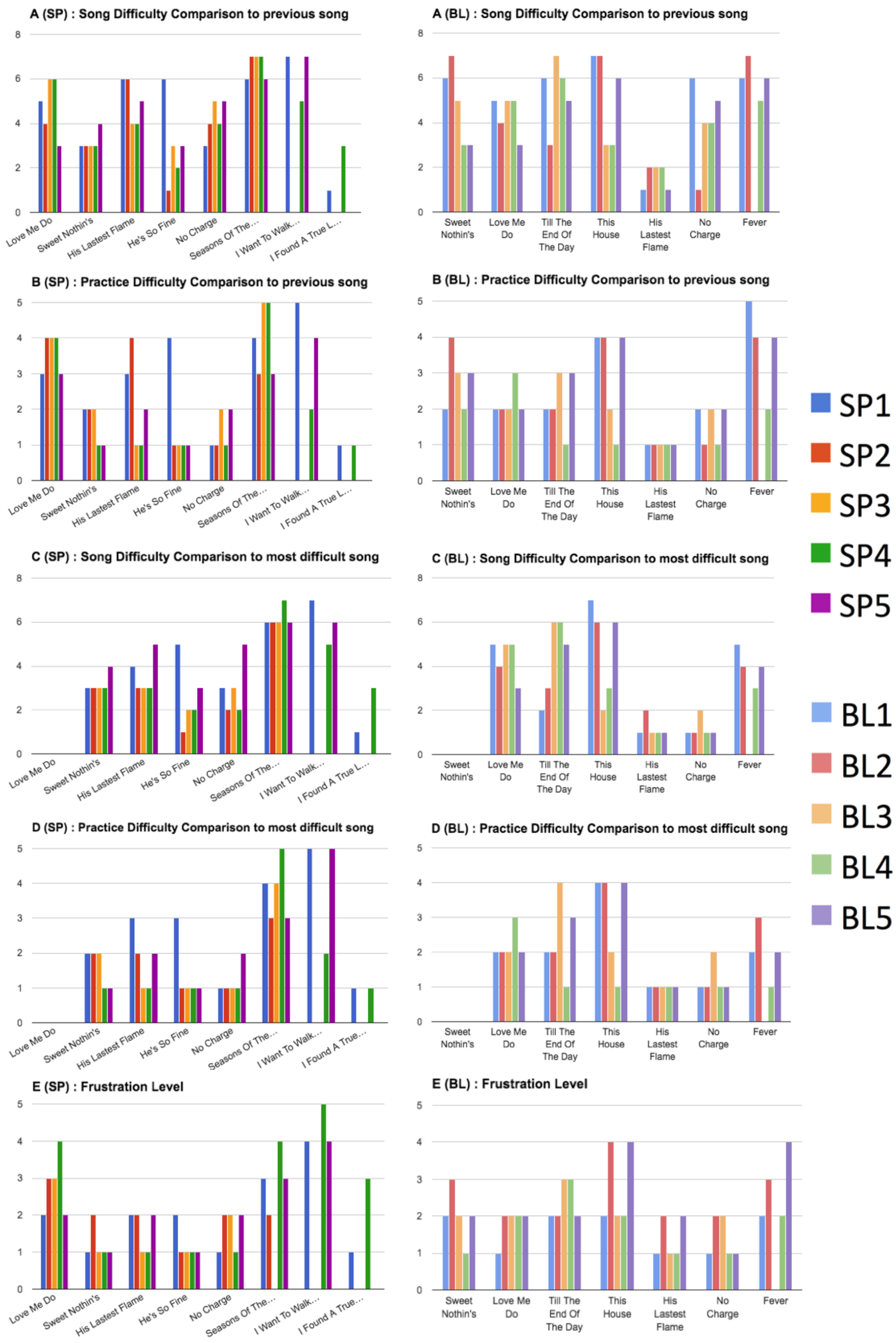

Figure 6.3 All results of the questionnaire responses. The horizontal line presents each song.

counted responses for up to that song, leading to 50 in total. The BL participants expressed 12 smooth and 13 hard perceived progression, respectively. These numbers for the SP participants were 15 and 10. A Chi-square test did not reveal a significant result ($\chi^2(1, N = 50) = 0.32$, $p = .57$, the odd ratio: 0.92). the response data implies that I did not observe a clear perceived progression difference in both conditions.

These results does not strongly suggest differences between the two conditions in terms of how smooth they felt the lesson progression was. I further looked into qualitative data to triangulate my finding.

6.5.3 Qualitative Results

Here I report representative qualitative results gained from the interviews with the participants. The quotes were translated into English as faithfully as possible.

Similar Effect by the Underlying Algorithms

In the baseline algorithm, participants were generally positive about the song order.

> *Well, I did not notice much in the order of the songs.* [BL2]

> *I think the song order was quite good.* [BL3]

However, I also observed that the BL participants often felt a large variance of the difficulty.

> *The difference between today's easy and difficult songs was really big. Last day was a difficult one, but today, [the lessons] became very easy, and then became very difficult... I really feel a big jump [in difficulty].* [BL1]

> *I suddenly had a chord that I need to hold with two fingers (a power chord). Before this time, it's like a mixture of what I have learned and new ones (chords). But this time, there were a lot of completely different ones (chords), which was a little hard.* [BL4]

This is an interesting observation as the baseline algorithm was intended to offer the easiest songs to practice first. However, as the participants mentioned, the song list generated by the baseline method did not necessarily provide smooth progression. This result clearly highlights limitations of this naïve method, encouraging future research on improving song prioritization algorithms.

I asked the same questions to the SP group and obtained similar impressions about the song order.

> *I think the song order is good. The songs get progressively longer. I think it's good.* [SP3]

> *It's like songs were getting gradually difficult, and I found it good.* [SP4]

But one of the SP participants commented that he felt a large leap in the seventh song.

> *The number of new chords suddenly increased [in the seventh song]. They also had some barre chords. If they appear one-by-one, I think I can do it, but they all appeared at once.* [SP1]

However, he followed up with acknowledging smooth progression offered by the SP algorithm.

> *Besides that (the seventh song), I had one or two new chords every song or every three lessons. I think it was the right amount.* [SP1]

Similar comments I obtained from both participants group may suggest that the algorithms did not affect the perceived difficulty progression much.

Positive Attitudes toward the Interface Design

The interviews also uncovered perceived benefits of the features in the Strummer interface. Participants agreed that step-by-step tutorials were beneficial. BL3 especially saw strong value in the chord transition practice stage.

> *I really like Transition (the chord transition practice stage). Simply thinking, if you can play individual chords, it's a matter of moving fingers fast enough. But, for example, movements are quite different between changing from G to C and from D to C. D and C are in similar locations, so I can move my index finger quickly. But for G, it's in a completely different location... Transition let me practice these, which I found really good.* [BL3]

In the step-by-step tutorials, the system included the chord recognition module to assess how well the participants played. This feature was also useful to the participants.

> *It was good that the system told me something I could not recognize by my ears... Well, for example, even when I did not place my fingers correctly, (the chord) still sounded OK to me. But if I looked at the screen, I failed. Then, I looked at my left hand, and found that my middle finger was in a wrong position.* [BL2]

Potential Discouragement by Scaffolding

As expected, barre chords were very difficult to play for most of the participants. Table 6.2 shows that half of the participants were not able to play the F and B chords even at the end of the study. Additional tutorials and scaffolding by synthesizing sounds are intended to support practicing such chords. However, the scaffolding often discouraged participants as SP1 and SP3 stated as follows:

> *I think it's good to have the assist (chord complement) mode of the system. However, I still want to have the performance judgment even when the assist mode is on. Because it motivates me.* [SP1]

> *I could not play F after all. I always had support for it. But that made me wonder if this is really OK. I felt like I would never become able to play it.* [SP3]

Barre chords are well known to be hard to play and an a discouraging element in guitar practice. Future work should investigate better support on practicing these chords.

Chord Performance Assessment Feedback

As mentioned above, the participants presented a positive reaction toward the performance assessment feature. However, some participants stated that the feedback was not sufficient to understand how to fix finger placement.

> *I tried to [my fix finger placement based on the feedback], but it is hard to understand.* [SP3]

However, BL1 stated an interesting opinion about the feedback as follows:

> *I think [the feedback on the played chord] is effective, maybe. When I felt like I missed something, the system also told me that I missed something. I think it actually help to build confidence even if we fail to play perfectly.* [BL1]

BL1's remark means that he judged his performance by listening to his playing without using the feedback. But he considered that the feedback was still useful to understand how well the chord has been played.

Learners have a different skill and experience even though they are novice. In the Strummer system, feedback design is fixed for all learners. Future work should investigate how adaptively feedback design can change depending on the learner's skill level and experience.

Chapter 7

Discussion

7.1 Effects of Prioritization Algorithms

My results suggest that there is not a large difference in perceived difficulty increase. Figure 7.1 shows a plot of the total chord difficulty in the two conditions. The baseline algorithm demonstrates a smooth slope while the SP method exhibits some fluctuations. Nevertheless, the quantitative results did not express major differences between the two methods. These pieces of evidence therefore are supportive of the main hypothesis.

7.2 Interface and Tutorial Design

The features in the Strummer interface were in general positively received. However, the scaffolding feature was not as effective as I intended. Future work should investigate other approaches to support practice on difficult chords, especially barre chords. The step-by-step tutorial design was seemingly beneficial. Future work can examine how a system can break down lessons more intelligently. One participant provided an interesting suggestion that lessons should be broken down so that learners can practice separate components of hand and finger movements.

> *If a lesson is like adding only one thing, such as moving a hand horizontally (changing the fret position), I can concentrate on it. Then, the next lesson would be only about barre chords so that I can get used to them. If the system separates these components, I think it might be easier.* [SP5]

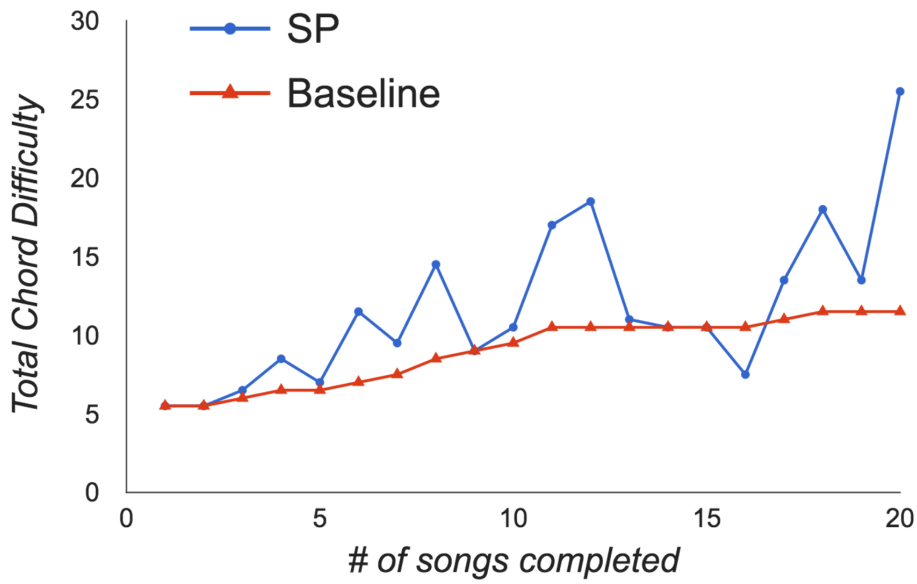

Figure 7.1 The total chord difficulty shift in the SP and baseline conditions.

I implemented Strummer as a system for guitar chord practice in this thesis. The system can be extended to other polyphonic musical instrument training. Developers can tailor the Strummer system for piano practice by updating the chord and transition difficulty scores and replacing diagrams. My chord transition difficulty model involves how fingers are moved from one chord to the other. This idea can be applied even for monophonic musical instruments. For example, a similar approach may quantify difficulty of playing a particular note and transition to another in the violin. This thesis investigates guitar chord practice, but contributes to a broader research space of musical instrument training.

7.3 Audio Analysis and Performance Assessment

My performance assessment module have some limitations, which could have impacted on user experience during the experiment. A false positive is a common issue. More specifically, the present implementation can not distinguish whether the sound is the fundamental tone or higher harmonics (overtones).

I explain this issue with the example of playing the C chord. The components of C chord in the Strummer system are $C3, E3, G3, C4, E4$ notes. The system matches the fundamental frequencies of a chord with the observed frequency peaks to determine whether it is correctly played. When learners play correctly, the system finds the sound peak in the fundamental frequencies. However, generated sounds also contain higher harmonics. When learners place fingers on the third fret of the fifth string and pluck, the guitar makes the $C3$ note prominently, which includes the fundamental frequency. At that time, higher harmonic frequencies of $C3$ can exhibit peaks. Especially, low-dimensional harmonics (2nd: $C4$, 3rd: $C5$, 4th: $G5$, etc.) may have a non-negligible power. Such sounds might overlap with other fundamental frequency peaks in the given chord (e.g., $C4$ on the second string). As a result, the system may incorrectly determine that learners have successfully played the C chord even if they fail to hold the second string.

To solve this problem, the system needs more powerful multipitch analysis. However, more sophisticated approaches can be computationally expensive and may not run in real time, which interferes with the interface. Future work should further examine how the chord recognition accuracy can be improved while maintaining its execution time.

Another issue is parameter setting. I did not implement automatic parameter tuning mechanisms in this work. I manually tuned the threshold value, and the system judges the frequency bin as a peak based on the value for each participant. Thus, the module does not perform well in the noisy environment. One possible improvement is to incorporate noise adjustment to adaptively change the threshold values. It would also be desirable that the module can judge how well learners play chords as well as whether correctly or not.

7.4 Limitations and Future Improvements

Besides the chord recognition module, there are limitations on the system and user study design.

Dataset

In this thesis, I only used the Billboard dataset for analysis. As previously mentioned in § 3.3, I gained top ten genres as shown in the Table 3.1. I discussed in § 3.5 that the trends of chord occurrences are different among genres. These trends should be characteristics of the given songs. If I assemble latest songs in the world, its trend would be different from the trend of the Billboard dataset. However, the results of my analysis affirmed that occurrences of chords and their transitions in given songs and genres follow the power law. This finding can be generalized into other datasets. When the given songs have a distribution of chord occurrences which fits a power law, the system can calculate legitimate *CP* and *SP*. Thus learners can start to practice from easy-to-learn chords in the given songs.

User Evaluation

In this thesis, I conducted a controlled study in the SP and BL conditions. This study clarified the benefits of the system as I previously described. However, my study did not show a significant difference in the participants' performance and impression between the SP and BL conditions. One reason is that my study was not long enough to compare potential of prioritization algorithms. The user study was five days long, and the participants practiced the songs listed in Table 6.1. The number of songs they practiced is less than ten, although the given song set in the user study has 115 songs. Consequently, participants only covered less than ten percent of entire given songs. To fully evaluate them, participants should complete practicing the given songs. If I conduct a longer study (e.g., one-month long), results could more clearly demonstrate the features of both algorithms. On the other hand, a long-term study makes the comparison difficult as I mentioned in § 6.1. An additional evaluation needs careful designing to keep a balance between the length of user study period and the given song set.

Another limitation which lies on the study design is the number of participants. I recruited ten people and divided them into two groups, but it was not a large number. A small number of participants result in large errors of statistics. If I gathered more people, results would be more reliable and could show significant differences.

Implementation

The present Strummer system uses the Billboard dataset that only contains a limited number of song. SP1 stated why he felt not highly motivated to practice as follows:

> *Well, I think the reason why I was not very motivated is that I couldn't see what chord and song I am playing. I think I will be more motivated if there are some songs I've heard.* [SP1]

To extend the dataset, the system may use other music data resource, including Songle has (Goto et al., 2011). Songle is a web service platform for active music listening. It automatically executes music structure recognition when users input the URL of an original song released in music web services (e.g., SoundCloud [1]). An external system can fetch musical annotations of desired songs in the JSON format through Songle Widget APIs (Goto

[1] https://soundcloud.com

et al., 2015). As mentioned in Section§ 3.4, the Strummer database is compatible with Songle Widget APIs, and thus can immediately import songs.

The present database only contain the chord information (Figure 3.1). Although this information is sufficient for chord practice purposes, additional data (e.g., melodies and lyrics) could enrich the Strummer tutorials. Similar to Songle (Goto et al., 2011), a future system could include video clips, for example.

In the current implementation, a web server runs in a local machine and thus I did not conduct any evaluation on a network delay. In the Strummer system, the performance assessment module should run in real time. If the communication between the web and client server generates a non-negligible delay, it will greatly degrade the user experience. Future work should investigate how to minimize such delays to make the system widely accessible.

Chapter 8

Conclusion

Many people have a desire to acquire musical instrument playing skills. Although recent online services offer a variety of training materials, the number of such resources is overwhelming, and learners may get lost in which to practice. I develop models and algorithms to identify important and easy-to-learn chords within a given music set. I instantiate them in Strummer, an interactive and multimedial system for guitar chord practice tailored toward novice learners. Strummer provides step-by-step tutorials to make practice manageable. A quantitative evaluation confirmed a potential of better skill transfer to unseen songs than the pure difficulty-based method. A user study revealed similar smooth lesson progression between my algorithm and baseline method.

List of Publications

Journal

Shunya Ariga, Masataka Goto, and Koji Yatani. An Interactive Guitar Chord Practice System with a Data-drive approach. In preparation for *IEICE Transaction on Information and Systems*.

International Conferences (peer-reviewed)

Shunya Ariga, Masataka Goto, and Koji Yatani. Strummer: an Interactive Guitar Chord Practice System. Submitted to *IEEE International Conference on Multimedia and Expo (ICME* 2017).

(Another submission about my side project in collaboration with Satoru Fukayama and Masataka Goto at AIST is currently in preparation for ISMIR 2017.)

Domestic Workshop (not peer-reviewed)

有賀 竣哉, 後藤 真孝, 矢谷 浩司, ‘‘Strummer:インタラクティブなギターコード演奏システム’’, 第 114 回音楽情報科学研究会(SIGMUS), February 2017.

References

Abramson, R. M. (1980). Dalcroze-based improvisations. *Music Educators Journal*, 66(5):62–68.

Barthet, M., Anglade, A., Fazekas, G., Kolozali, S., and Macrae, R. (2011). Music recommendation for music learning: Hotttabs, a multimedia guitar tutor. In *Workshop on Music Recommendation and Discovery*, pages 7–13.

Brown, P. F., deSouza, P. V., Mercer, R. L., Pietra, V. J. D., and Lai, J. C. (1992). Class-based n-gram models of natural language. *Comput. Linguist.*, 18(4):467–479.

Burgoyne, J. A., Wild, J., and Fujinaga, I. (2011). An expert ground truth set for audio chord recognition and music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, ISMIR '11, pages 633–638.

Dannenberg, R. B., Sanchez, M., Joseph, A., Capell, P., Joseph, R., and Saul, R. (1990). A computer-based multi-media tutor for beginning piano students. *Journal of New Music Research*, 19(2–3):155–173.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

Dreyfus, H. L., Dreyfus, S. E., and Athanasiou, T. (1986). *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. The Free Press.

Dreyfus, S. E. and Dreyfus, H. L. (1980). A five-stage model of the mental activities involved in directed skill acquisition. Technical Report ORC-80-2, University of California, Berkeley Operations Research Center.

Feygin, D., Keehner, M., and Tendick, F. (2002). Haptic guidance: experimental evaluation of a haptic training method for a perceptual motor skill. In *Proceedings 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems. HAPTICS 2002*, pages 40–47.

Fitts, P. M. and Posner, M. I. (1967). *Human performance*. Brooks/Cole.

Glise, A. (1997). *Classical Guitar Pedagogy: A Handbook for Teachers*. Mel Bay Publications, Inc.

Goto, M., Yoshii, K., Fujihara, H., Mauch, M., and Nakano, T. (2011). Songle: A web service for active music listening improved by user contributions. In *ISMIR*, pages 311–316.

Goto, M., Yoshii, K., and Nakano, T. (2015). Songle widget: Making animation and physical devices synchronized with music videos on the web. In *IEEE International Symposium on Multimedia (ISM)*, pages 85–88.

Grindlay, G. (2008). Haptic guidance benefits musical motor learning. In *2008 Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 397–404.

Harte, C., Sandler, M. B., Abdallah, S. A., and Gómez, E. (2005). Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 6th International Society for Music Information Retrieval Conference*, ISMIR '05, pages 66–71.

Huang, K., Do, E. Y.-L., and Starner, T. (2008). Pianotouch: A wearable haptic piano instruction system for passive learning of piano skills. In *Proceedings of the 2008 12th IEEE International Symposium on Wearable Computers*, ISWC '08, pages 41–44.

Jaques-Dalcroze, E. (1930). Eurhythmics art and education (1930).

Johnson, R., van der Linden, J., and Rogers, Y. (2011). Real-time feedback for learning the violin. In *Proceedings of the 25th BCS Conference on Human-Computer Interaction*, BCS-HCI '11, pages 535–538.

Kurosawa, Y. and Suzuki, K. (2010). Robot-assisted playing with fingering support for A saxophone. In *Proceedings of the 2010 International Computer Music Conference*, ICMC '10.

Lidy, T., Rauber, A., Pertusa, A., and Iñesta, J. M. (2007). Improving genre classification by combination of audio and symbolic descriptors using a transcription systems. In *Proceedings of the 8th International Society for Music Information Retrieval Conference*, ISMIR '07, pages 61–66.

Mauch, M. and Dixon, S. (2012). A corpus-based study of rhythm patterns. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, ISMIR '12, pages 163–168.

McPherson, A. P., Gierakowski, A., and Stark, A. M. (2013). The space between the notes: Adding expressive pitch control to the piano keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2195–2204.

Motokawa, Y. and Saito, H. (2006). Support system for guitar playing using augmented reality display. In *Proceedings of the 5th IEEE and ACM International Symposium on Mixed and Augmented Reality*, ISMAR '06, pages 243–244.

Nanni, L., Costa, Y. M., Lumini, A., Kim, M. Y., and Baek, S. R. (2016). Combining visual and acoustic features for music genre classification. *Expert Systems with Applications*, 45:108 – 117.

Ng, K. and Nesi, P. (2008). i-maestro : Technology-enhanced learning and teaching for music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 225–228.

Ng, K. C., Weyde, T., Larkin, O., Neubarth, K., Koerselman, T., and Ong, B. (2007). 3d augmented mirror: A multimodal interface for string instrument learning and teaching with gesture support. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, ICMI '07, pages 339–345.

Ogihara, M. and Li, T. (2008). N-gram chord profiles for composer style representation. In *Proceedings of the 9th International Society for Music Information Retrieval Conference*, ISMIR '08, pages 671–676.

Ojima, Y., Nakamura, E., Itoyama, K., and Yoshii, K. (2016). A hierarchical bayesian model of chords, pitches, and spectrograms for multipitch analysis. In *International Society for Music Information Retrieval Conference*, ISMIR '16, pages 309–315.

Pachet, F., Roy, P., Moreira, J., and d'Inverno, M. (2013). Reflexive loopers for solo musical improvisation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 2205–2208.

Percival, G., Wang, Y., and Tzanetakis, G. (2007). Effective use of multimedia for computer-assisted musical instrument tutoring. In *Proceedings of the International Workshop on Educational Multimedia and Multimedia Education*, Emme '07, pages 67–76.

Pérez-Sancho, C., Rizo, D., and Iñesta, J. M. (2009). Genre classification using chords and stochastic language models. *Connection Science*, 21(2&3):145–159.

Takegawa, Y., Tsukamoto, M., and Terada, T. (2011). Design and implementation of a piano practice support system using a real-time fingering recognition technique. In *Proceedings of the 2011 International Computer Music Conference*, ICMC '11, pages 387–394.

Tzanetakis, G. and Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302.

van der Linden, J., Johnson, R., Bird, J., Rogers, Y., and Schoonderwaldt, E. (2011). Buzzing to play: Lessons learned from an in the wild study of real-time vibrotactile feedback. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 533–542.

Yuksel, B. F., Oleson, K. B., Harrison, L., Peck, E. M., Afergan, D., Chang, R., and Jacob, R. J. (2016). Learn piano with bach: An adaptive learning interface that adjusts task difficulty based on brain state. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 5372–5384.

Appendix A

Guitar Chords Diagrams and Difficulties

The following figures present all chord diagrams used in the Strummer system. The number beside chord name represents its difficulty.

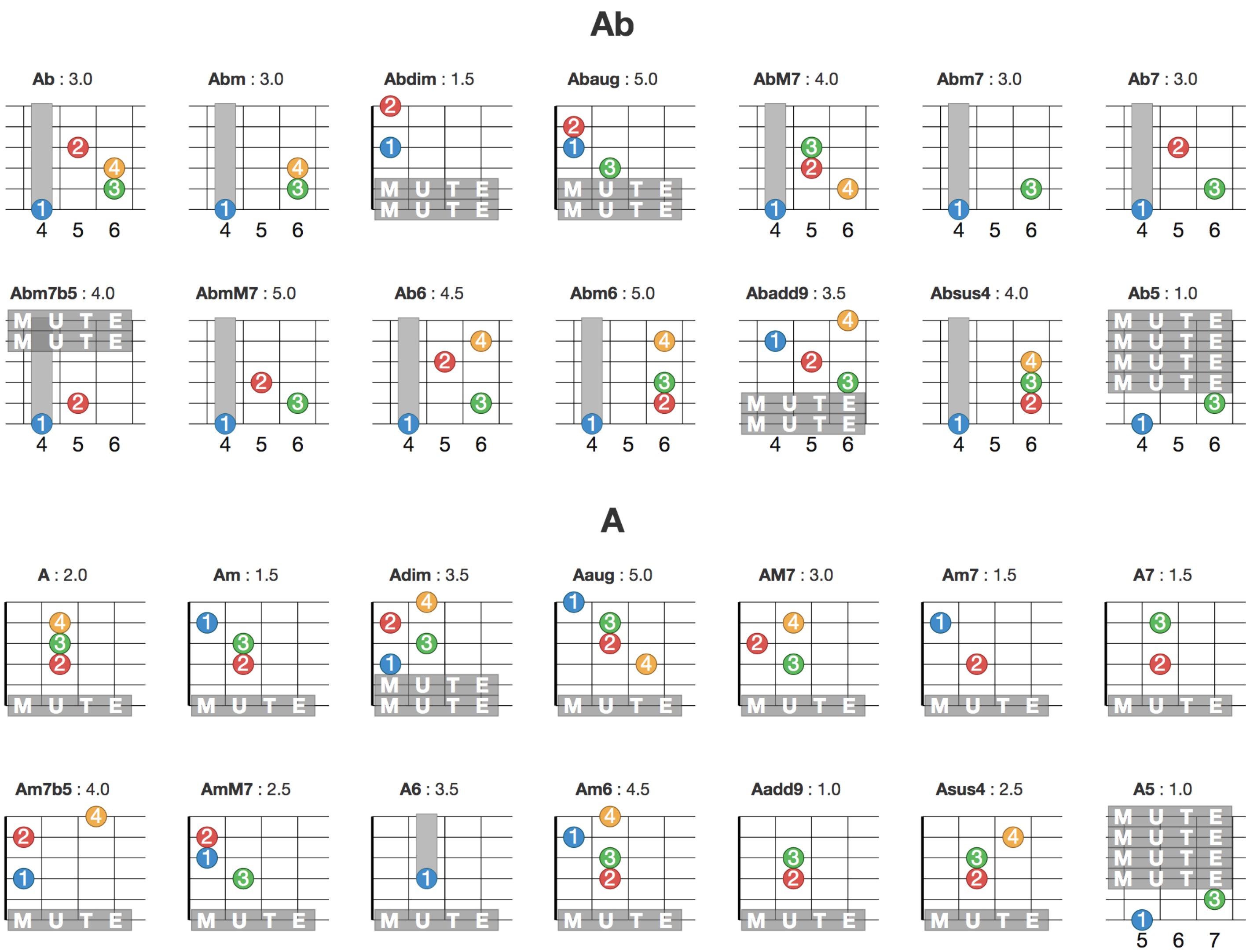

Figure A.1 All chord diagrams of root *Ab* and *A*.

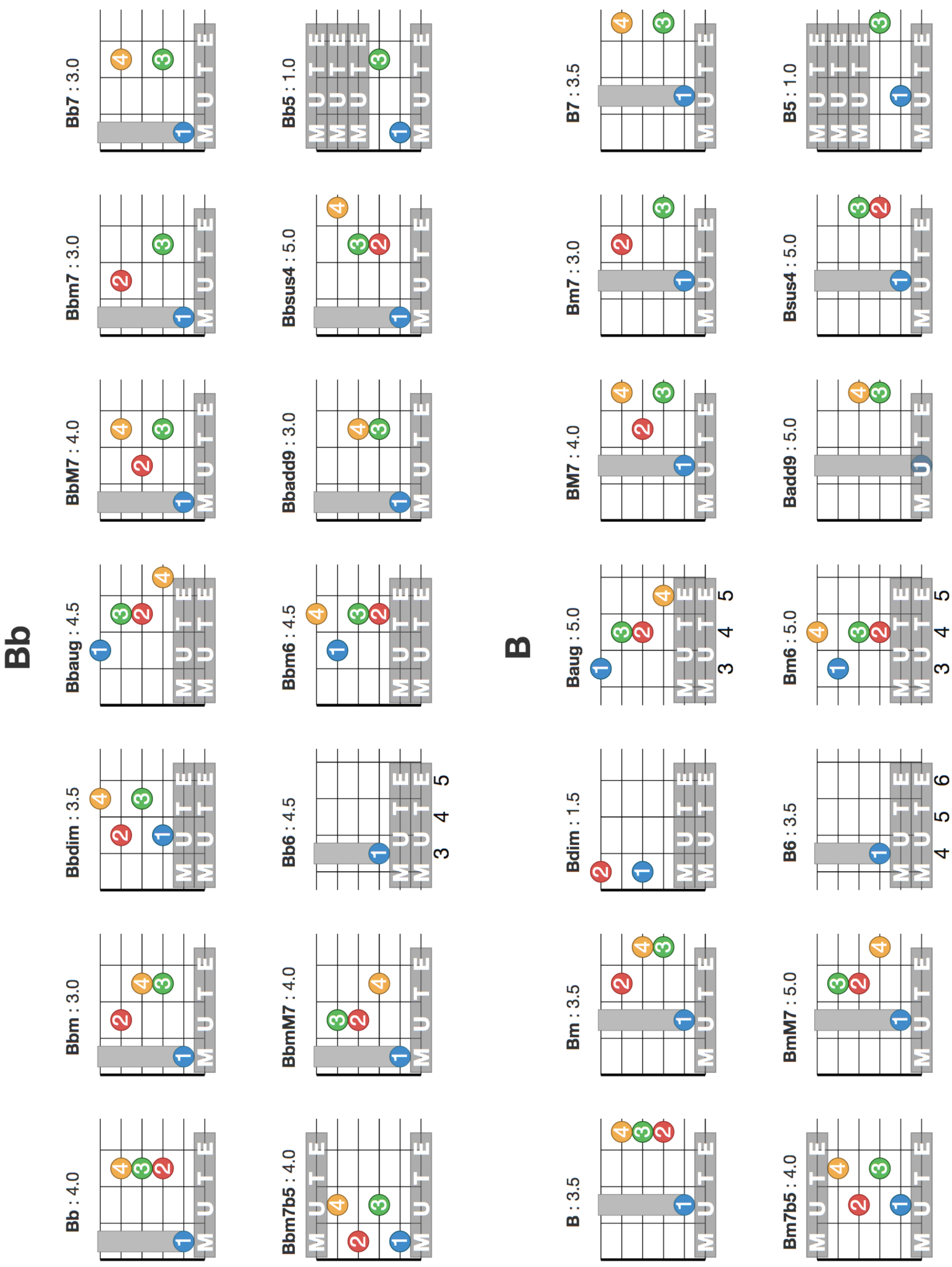

Figure A.2 All chord diagrams of root *Bb* and *B*.

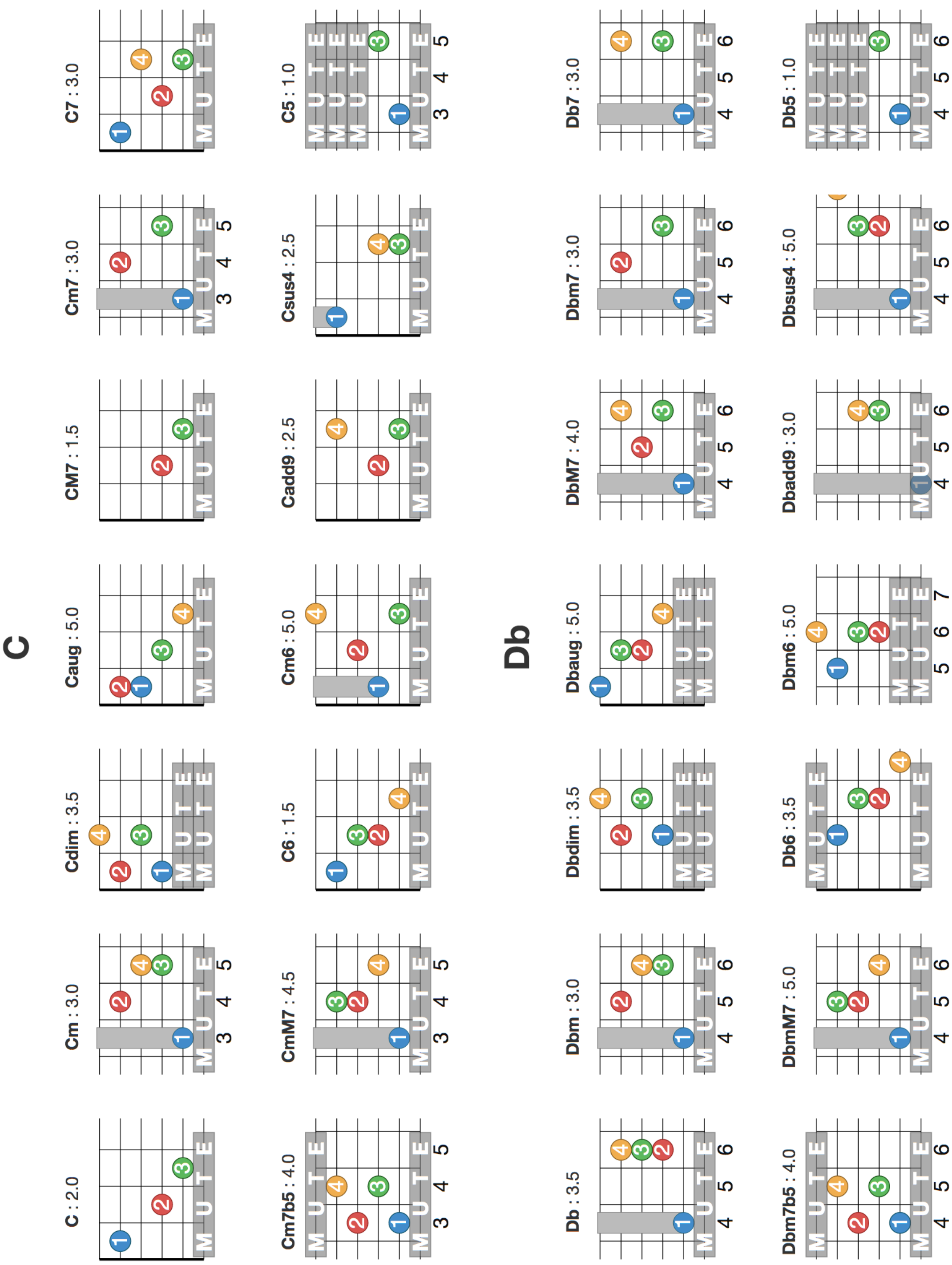

Figure A.3 All chord diagrams of root *C* and *Db*.

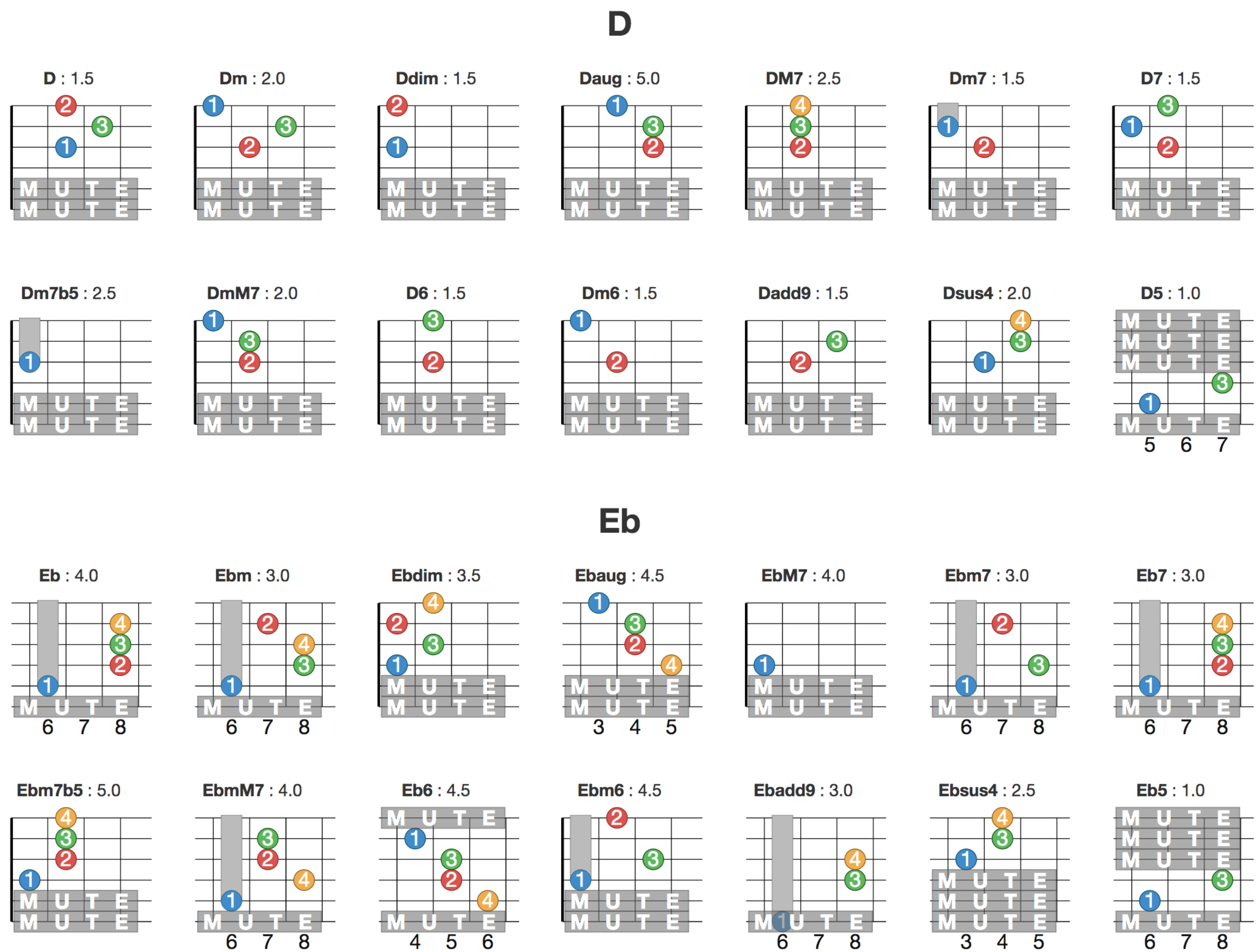

Figure A.4 All chord diagrams of root *D* and *Eb*.

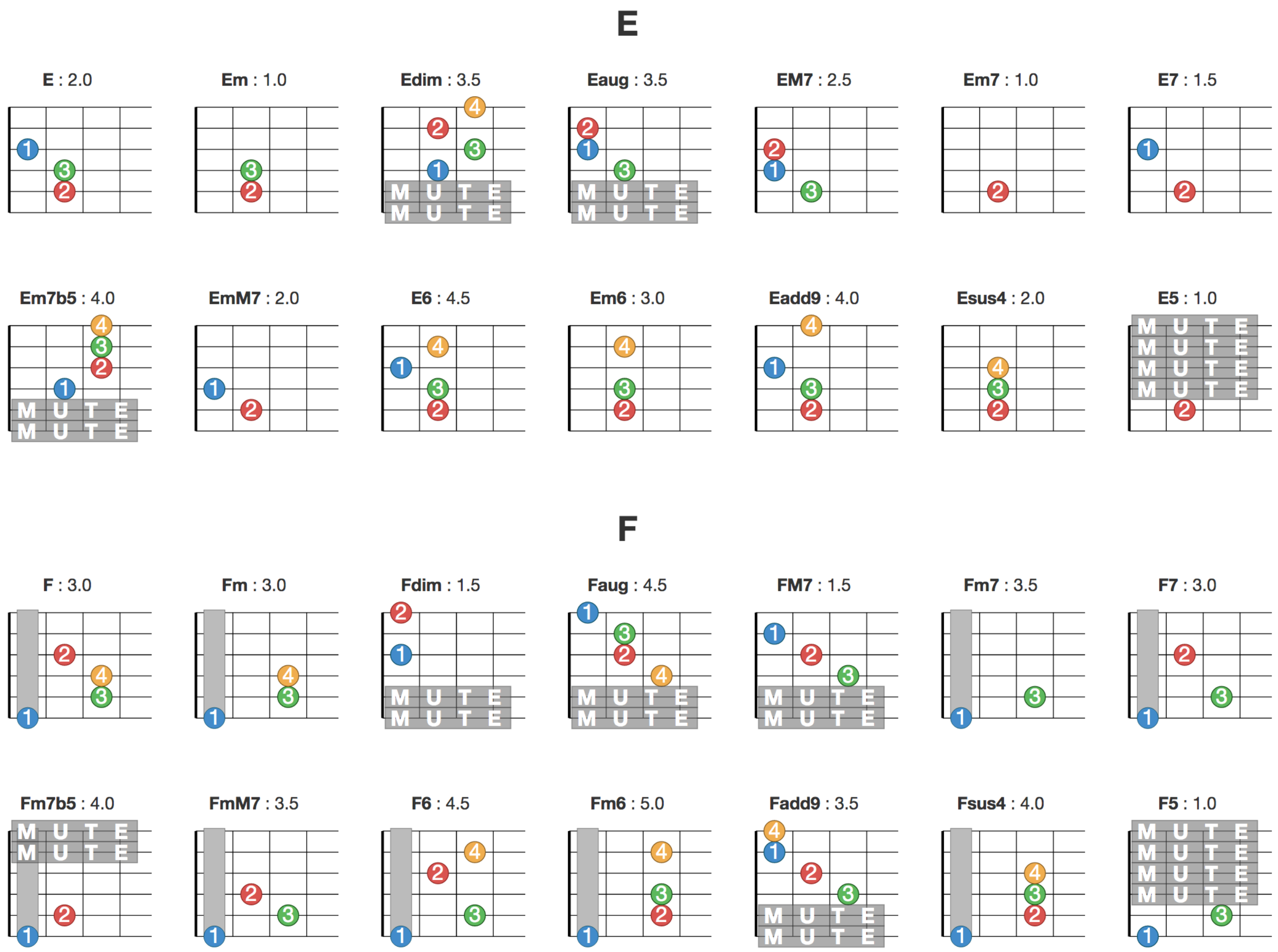

Figure A.5 All chord diagrams of root E and F.

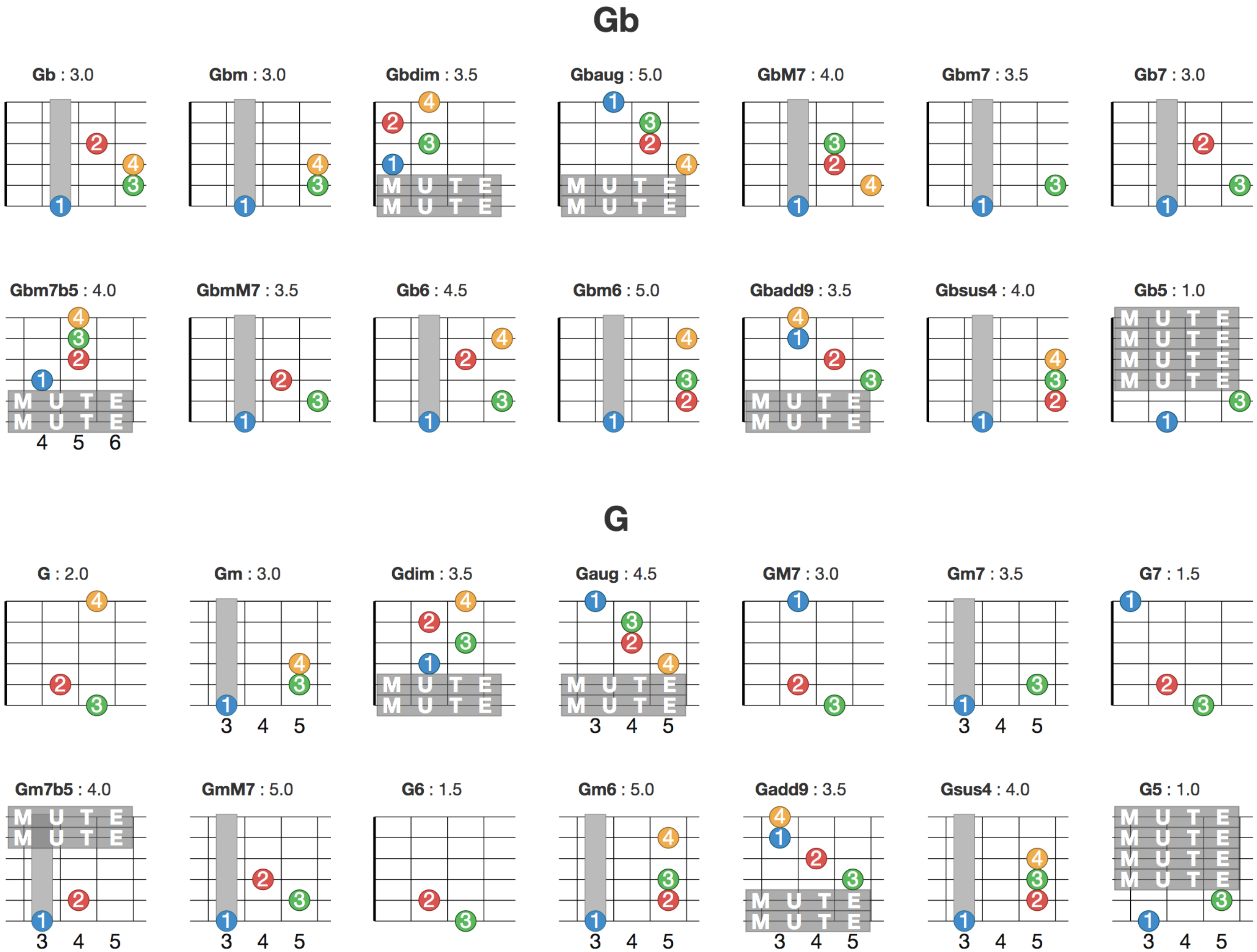

Figure A.6 All chord diagrams of root *Gb* and *G*.