A Linear-Chain Conditional Random Field Approach to the Guitar Fingering Problem

Master-Thesis von Michael Rau
Tag der Einreichung:

1. Gutachten: Prof. Dr. Johannes Fürnkranz
2. Gutachten: Dr. Frederik Janssen

TECHNISCHE UNIVERSITÄT DARMSTADT

Department of Computer Science
Knowledge Engineering Group

A Linear-Chain Conditional Random Field Approach to the Guitar Fingering Problem

Vorgelegte Master-Thesis von Michael Rau

1. Gutachten: Prof. Dr. Johannes Fürnkranz
2. Gutachten: Dr. Frederik Janssen

Tag der Einreichung:

Erklärung zur Master-Thesis

Hiermit versichere ich, die vorliegende Master-Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 22. März 2016

(Michael Rau)

Abstract

The guitar fingering problem involves the automatic generation of left-hand fingerings for guitar pieces.

This thesis covers the extension of solutions which map the problem to finding an optimal path in a graph. The presented approach is based on a linear-chain conditional random field (CRF) and can generate fingerings to any guitar piece regardless of its note structure. It is the first approach to incorporate knowledge on the physical dimensions of guitars and guitarists.

The generated fingerings were mostly appropriate regarding the choice of fretboard positions whereas the quality of the finger assignments was lacking. Compared to measuring distances on the guitar by a number of frets, the inclusion of the physical dimensions led to a slightly more accurate choice of fretboard positions. Major improvements are expected once higher-order dependencies between more than two subsequent notes are modeled.

Contents

List of Figures

List of Tables

1 Introduction

1.1 The Guitar Fingering Problem

Sheet music for the guitar specifies the notes of a piece with the same musical notation used for other instruments such as the piano. Unlike the piano, where a one-to-one mapping exists between the notes of a piece and the keys on the keyboard, guitars offer multiple locations on the fretboard to play one note. A frequently mentioned example of this specialty [19–22, 24, 31, 33] is the note E4 which appears on five different locations on the fretboard, as depicted in figure 1.

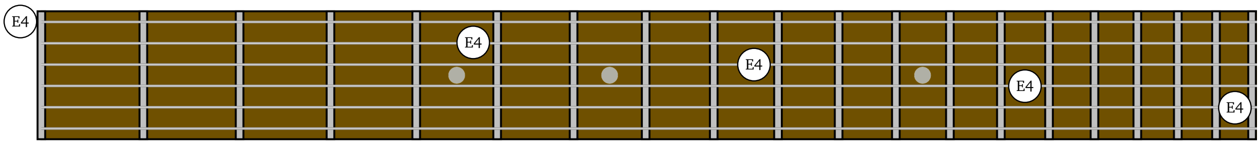


Figure 1.: The five locations of E4 on the fretboard of a classical guitar with standard tuning and 19 frets (adapted from Tuohy et al. [33, figure 1])[1)]

To assist guitarists in playing a piece, the musical notation is commonly supplemented by a *fingering*. A fingering assigns a finger of the left hand and a position on the fretboard, defined by a string and a fret, to notes of a guitar piece.

However, sheet music published for the guitar typically provides this assistance only for difficult sections or sections where the publisher deems it to be necessary. It is left out for sections where the fingering is considered to be obvious to the performer. While this approach ensures tidy sheet music it hinders lesser skilled guitarists, who lack the experience to complete the omitted fingering, at learning and playing a piece. Depending on the guitar skills of the editor, fingerings may be of poor quality or cumbersome to others, forcing guitarists to create their own fingerings. Since the creation of a good fingering is a complex task (more on this later) a system which automatically completes or generates fingerings of guitar pieces would be of great use to guitarists.

The task of generating a fingering for a guitar piece is commonly known as the guitar fingering problem [23]. The formal definition is: Given a guitar piece, assign a string, a fret and a finger to each note so that the fingering is optimal according to a quality measure.[2)]

An optimal fingering to a piece should be easy to play: It should avoid large jumps of the left hand in time sensitive sections, should favor comfortable finger positions and should minimize the amount of strength needed to play. The memorability of a fingering should be ensured by reusing finger positions within the piece and across all pieces known by the performer. Most importantly, it should recreate the feeling and acoustic outcome intended by the composer and the performer.

The computational complexity of the guitar fingering problem arises from the multitude of locations on the fretboard available for each note: In the abovementioned example of E4, four of the five locations can be played by either of the four fingers of the left hand commonly used for playing the guitar. The remaining location requires no action of the left hand to be played. This leaves one with 17 different

1) Released under CC BY-SA 3.0. Fretboard inlay by Wikimedia user GreyCat, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=520147

2) Finding a fingering does not include altering notes of a piece. This is part of another task called arranging, where notes of a piece written for a certain instrument are omitted or changed to render it playable on a different instrument. Several attempts have been made at automatically arranging pieces for the guitar [9–11, 30, 32], however, this thesis focuses solely on generating fingerings.

possibilities of playing the same note. As a result, a melody consisting of n notes can feature up to 17^n different fingerings in the worst case scenario, ruling out exhaustive search as a trivial solution approach. Allowing polyphonic pieces increases the complexity of the task because one needs to pay attention to additional dependencies among notes in a chord, i. e. whether a guitarist can play several positions on the fretboard at the same time regarding biomechanical aspects of the human hand.

To make things worse, optimal fingerings depend on emotions which are subjective by nature, meaning an optimal fingering is (in part) a matter of opinion. Guitarists might prefer a different, but equally difficult fingering over one found in published sheet music because they are of the opinion that a certain note fits the mood of the piece better when played on a string of a different timbre [33, p. 4]. Put in other words, is is likely that more than one "optimal" fingering exists for a piece.

In summary, the guitar fingering problem is about finding one among several good solutions in a large search space.

1.2 Thesis Structure

The following chapter provides an introduction to the guitar as an instrument and various aspects of difficulty associated with playing the guitar. The literature review in chapter 3 offers an insight into existing solutions to the guitar fingering problem.

An introduction to linear-chain conditional random fields is provided in chapter 4, followed by chapter chapter 5 explaining their application to the guitar fingering problem. Chapter 6 touches upon the implementation of the presented approach and reports the selection process for the cost function components of the linear-chain conditional random field.

In chapter 7, several generated fingerings are presented and evaluated. A discussion of the results follows in chapter 8. The thesis closes with a summary and suggestions for further research on the topic in chapter 9.

2 The Classical Guitar

This chapter provides knowledge on the classical guitar which is necessary to follow the reasoning of this thesis and reinforces the points of the motivation.

The relevant parts of a classical guitar are explained first, followed by guitar playing techniques. An overview of common notations encountered for guitar pieces and fingerings is presented afterwards. Finally, the difficulty aspects of guitar playing are stated.

2.1 The Classical Guitar in its Parts

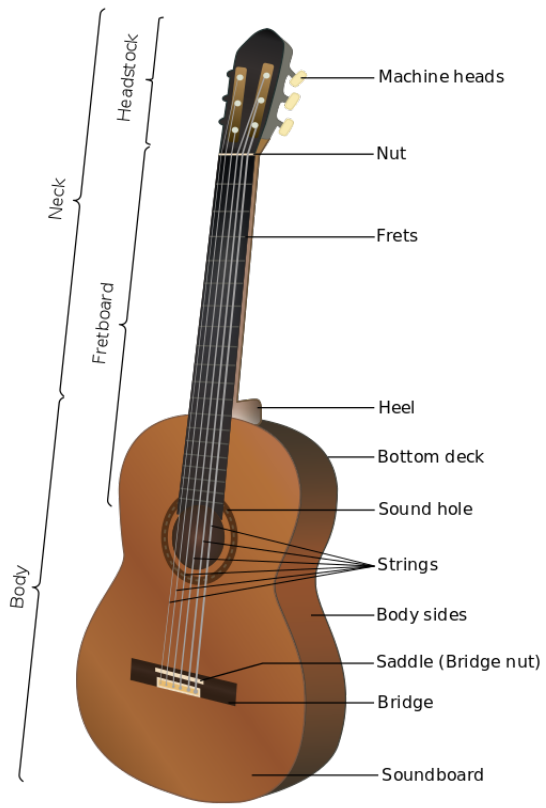


Figure 2.: Parts of a classical guitar[1)]

A classical guitar (as shown in figure 2) features 6 strings whose pitch is specified by a tuning. The standard tuning is E2, A2, D3, G3, B3, E4 [27, p. 5]. Note that the intervals between the strings are perfect fourths (5 semitones) except for the interval between the G and B string, which is a major third (4 semitones). Each guitar string can also be identified independently of its pitch: The standard numbering scheme for strings associates strings with numbers ascending from 1 to 6, where 1 is assigned to the string of the highest pitch (the rightmost string in figure 2) [27, p. 5]. The vibrating part of a guitar string is bounded by the saddle on the corpus (or body) and the nut. The distance between these two reference points is called *scale length* [14]. Classical guitars are built with a scale length of 650 mm. The distance between the strings and the fretboard (the *action*) increases towards the saddle.

1) This image was created by William Crochot. William Crochot – Acoustic guitar parts.png, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=37646236

Depressing a string midway between nut and saddle doubles its vibration frequency when plucked, producing a pitch which is one octave (12 semitones) higher than the pitch the string is tuned to. Guitars are fretted instruments which means pitch increments on the guitar are discretized into semitones by frets (small metal strips on the fretboard). For this reason, 12 frets are distributed over the area between the nut and the midway point of the string. The spacing between the frets needs to be non-uniform to achieve a clean intonation. Given the scale length l of a guitar, the distance d between the nut and fret fr is defined by

$$d = l \cdot \left(1 - 2^{-\frac{fr}{12}}\right)$$

according to Mottola [14].

The frets of a guitar are identified by the number of semitones they add to the base pitch of a string, meaning the fret at the midway point receives number 12[2)]. The highest fret found on a classical guitar is fret 19. Figure 3 shows the mapping from fretboard positions to pitches for a guitar with standard tuning. Notice how the tuning affects that any pitch on string s appears four or five frets higher on the string $s+1$.

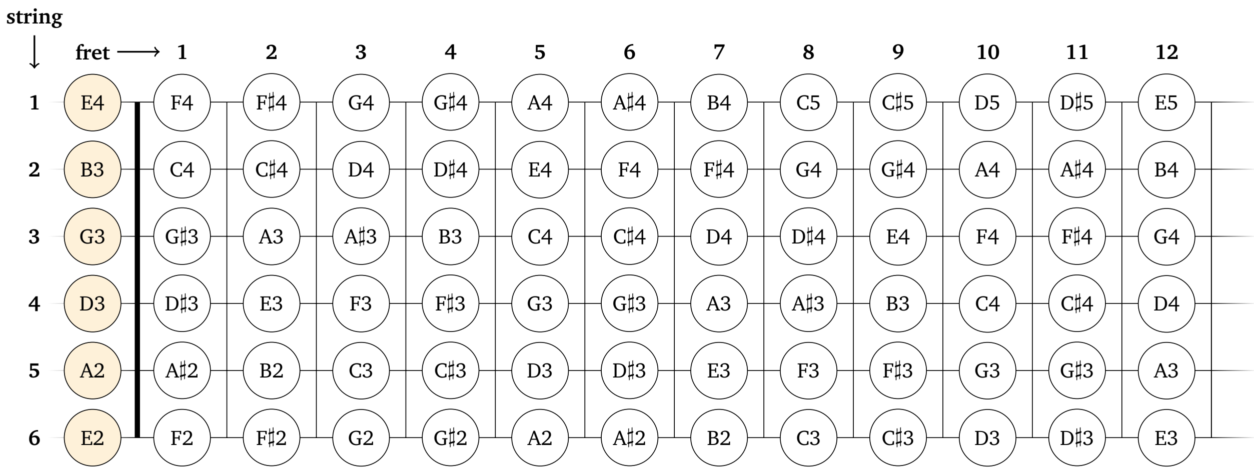


Figure 3.: Pitches produced by fretboard positions up to the 12th fret on a guitar with standard tuning

2.2 Playing Technique

A guitarist performs different tasks with each hand when playing the guitar: The fingers of the left hand depress strings on the fretboard while the fingers of the right hand produce the sound by strumming or plucking the strings[3)]. The classical guitar playing technique incorporates no additional equipment such as plectrums commonly used for playing the electric guitar.

All fingers on the left hand except for the thumb may depress strings. The fingers are identified by the numbers 1 to 4, starting with the index finger [27, p. 16]. The term *open string* refers to a string which is played without a finger of the left hand depressing it. The position of the left hand along the fretboard is referred to by the fret in which the index finger is located. In case the index finger does not depress any strings, its fret is estimated under the assumption that the fingers of the left hand are located in consecutive frets.

A complete left-hand fingering of a guitar piece assigns a string, a fret and a finger to each note. The number 0 is used for the finger assignment of open strings [27, p. 16]. Left-hand fingerings stand opposed

2) Frets are usually written in roman numerals – arabic numerals are used throughout this thesis instead to avoid confusion in calculations.

3) The roles may be inverted for a left handed guitarist. Nonetheless, the term *left hand* is used synonymously for the hand depressing strings on the fretboard.

to right-hand fingerings which specify a finger of the right hand for each note of a piece. Right-hand fingerings are not as relevant as left-hand fingerings to this thesis, therefore their details are not covered in this introduction. The term *fingering* is used as a synonym to left-hand fingerings throughout this thesis.

2.3 Notation Systems for Guitar Pieces and Fingerings

Sheet music for the guitar uses standard musical notation. Musical notation can incorporate fingering information, however, it becomes crowded if a string, a fret and a finger were indicated for each note. As a result, fingering information is only provided for certain sections or is omitted entirely.

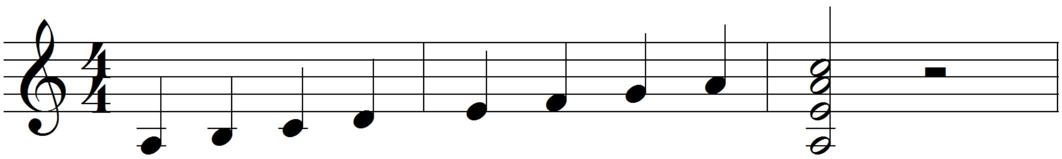
Alternative notations for fingerings exist in the form of tablature and fretboard charts.

Tablature represents each string of a guitar by a horizontal line. To notate a note played on string s in fret fr, the number fr is written on the line corresponding to string s. Finger assignments are commonly written below the horizontal lines. The temporal order of the notes is indicated by the horizontal arrangement of the fret numbers. Chords are notated by a stack of fret numbers. The finger assignments of chords are likewise notated by a stack of numbers which replicates the order of the fret numbers.

Tablature is only intended as a supplement to musical notation and as such provides no temporal information except for the relative order of the notes and bar lines. It is commonly shared online in textual representation (*ASCII tablature*) due to its portability compared to sheet music.

A fretboard chart visualizes the fretboard positions and finger assignments of a single chord on a stylized fretboard. It can be understood as a vertical slice taken from guitar tablature. Strings which are not supposed to be played are marked by an × symbol.

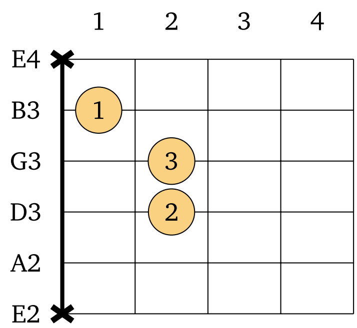
An example for each notation is shown in figure 4.



(a) Musical notation of an A minor scale and an A minor chord

```
E4|--------------|--------------|--------------|
B3|--------------|--------------|----1---------|
G3|--------------|--------0--2--|----2---------|
D3|-----------0--|--2--3--------|----2---------|
A2|--0--2--3-----|--------------|----0---------|
E2|--------------|--------------|--------------|
        2  3        2  3     3       1
                                     3
                                     2
```

(b) ASCII tablature showing a possible fingering



(c) Fretboard chart of the chord fingering in the third bar

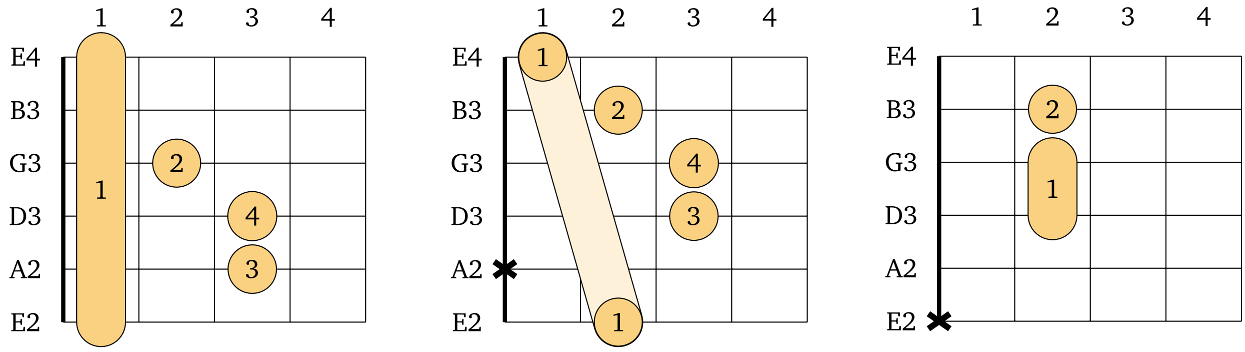
Figure 4.: Common types of notation for guitar pieces

2.4 Advanced Playing Techniques

The technique of depressing multiple strings with the same finger is referred to as a *barre chord*. The most common type of barre chords are index finger barre chords, where the extended index finger is placed flat over multiple strings (see figure 5a).

Several rarer variants of barre chords exist, such as barre chords played with different fingers than the index finger. It is also possible to play positions in different frets with one finger. This requires a diagonal positioning of the finger relative to the frets, hence the name diagonal barre chord (see figure 5b).

In a partially depressed barre chord, a finger depresses multiple strings in the same fret but is raised to accommodate space for other fingers or to allow for playing higher strings openly (see figure 5c).



(a) F major barre chord **(b)** F♯ major 7th diagonal barre chord **(c)** A major partially depressed barre chord

Figure 5.: Fretboard charts of barre chords

2.5 Difficulty Aspects of Guitar Playing

The challenge of playing an instrument has been shown to be of both biomechanical and cognitive nature [7].

The biomechanical challenges of playing the guitar mainly concern actions of the left hand. The left hand and its fingers need to be moved to their target positions on the fretboard in time before the right hand plays the respective strings. Fingers of the left hand endure potentially uncomfortable joint angles whilst exerting enough force to keep strings from producing an undesirable buzzing sound. A lot of force is necessary, in particular for playing barre chords and for playing in higher frets due to the increased distance between strings and fretboard. Plucking or strumming strings with the right hand requires less force. Also, the right hand is held in the same position near the sound hole most of the times, occasionally moving along the strings to change the timbre of the sound. These movements are slower, cannot be missed and are overall less critical to the rendition of a piece than the movements of the left hand.

Cognitive efforts of guitar playing involve the retrieval of fingerings from memory and the temporal coordination between actions of both hands.

Experience and sufficient practice of a piece diminish the perceived difficulty of these aspects.

A behavioral study conducted by Heijink et al. [7] on the motor movements of professional guitarists provides valuable insights to guitar playing. One finding was that guitarists preferred playing in lower frets over playing in higher frets, presumably because of the familiarity of these positions. They also documented that guitarists, when prompted to play at a fixed tempo, play notes leading up to difficult sections ahead of time to compensate for an expected time loss.

3 Review of Previous Approaches

Previously published approaches to the guitar fingering problem are reviewed in this chapter. Most of the approaches belong to one of two families of approaches, characterized by their representation of the optimization problem: Path-based approaches map the task of finding an optimal fingering to finding an optimal path in a graph. Evolution-based approaches start with one or more arbitrary fingerings which are gradually refined until no more improvements are possible.

Path-based approaches are reviewed first, followed by evolution-based approaches. Other notable approaches not belonging to the two families are presented afterwards.

3.1 Path-Based Approaches

The basic concept

The idea of casting the guitar fingering problem as a path optimization problem appears as early as 1989 in a paper by Sayegh [25] in which he introduces the "optimum path paradigm", a basic concept common to all path-based approaches:

Given a melody, one can create a layered (directed acyclic) graph with as many layers as there are notes in the melody. Each layer contains nodes representing the various positions on the fretboard where a certain note can be played. Based on the assumption that the difficulty of transitioning between fingerings of neighboring layers only depends on the two fingerings directly involved, one may define transition costs for each pair of nodes taken from two neighboring layers. At this point, finding the optimal fingering to the melody is identical to finding the path covering all layers with minimum costs. This can be done efficiently by the help of the Viterbi algorithm [25] (a dynamic programming algorithm which is explained in section 4.2). The transition costs between fingerings can be calculated by a cost function instead of declaring them for each individual pair [25].

The independence assumption (called *adjacent note assumption* by Rutherford [24]) is a key simplification of the problem domain necessary for efficient inference in path-based approaches. However, the gained efficiency comes at the expense of fingering quality: A cost function defined for the transition between two notes cannot incorporate knowledge on the fretboard positions of previous notes, meaning higher order concepts of a good fingering such as reoccurring patterns in the fingering cannot be implemented.

Introducing polyphony

Sayegh's work was improved upon by Radisavljevic et al. [23] who add support for basic polyphony. Their cost function is a linear term of several features (of which few are mentioned) characterizing aspects of fingering transitions and a new type of static costs used to penalize certain chord fingerings. The need to restrict or penalize fingerings arises from the fact that not every combination of multiple fretboard positions is biomechanically possible to play for the human hand. The feature weights were then optimized by a technique called "path difference learning" to fit published tablature of seven guitar pieces. Radisavljevic et al. report that their approach adapts well to the training data but performs poorly on a test set. Neither an evaluation measure nor the composition of the test set is stated.

Radicioni et al. [20] cover the fingering of chords in greater detail. Whereas Radisavljevic et al. [23] filter out unplayable chord fingerings by means of their cost function during the inference phase (thereby examining many meaningless states), Radicioni et al. aim to keep the number of nodes in the layered graph low by adding only the playable fingerings of a chord in the first place. The playable chord fingerings are generated by enumerating the solutions to a constraint satisfaction problem (CSP). A CSP is specified by several variables, their domains and constraints between those variables. Radicioni et al.

model the fingering of a single note as a 3-tuple of integer variables: one variable for the string, one for the fret and another representing the finger that plays the position. Constraints enforce the principles of guitar playing and the biomechanical properties of a performer's hand. The quality of the fingerings was judged by guitar experts who asserted that every generated fingering is viable. Compared to a set of fingerings created by the experts themselves the CSP approach missed 2.75 % of the fingerings found by the experts.

Cost function considerations

In another paper, Radicioni et al. [21] explain their take on the cost function for fingering transitions. Based on the research of Heijink et al. [7] a cost function was established which judges both hand movement along the fretboard and finger movement across the fretboard. Costs are defined between pairs of fingered positions (i. e. between 3-tuples of string, fret and finger), hence for calculating the total cost of a transition between two fingerings, the sum of the costs for each pair of fingered positions is calculated.

Movement along the fretboard is expressed by a term for the fretwise distance between the two fingered positions and a term which penalizes playing at higher frets. The distance term is discounted for hand movements towards the corpus. The costs for movement across the fretboard depend on the number of strings between the two fingered positions. To reward comfortable fingerings, Radicioni et al. additionally employ the concept of a comfort span for fretwise and stringwise distances between fingered positions. They emphasize however, that all parameters, costs and comfort spans were determined by manual experiments (without stating on which data the experiments were conducted).

The system was evaluated on three guitar pieces by comparing generated fingerings to fingerings created by a guitar expert, counting the matching fingered position 3-tuples as the quality measure. Across all pieces, 90.61 % of 3-tuples matched those from the reference fingering. The guitar expert was also asked to provide feedback to the generated fingering. He criticized certain sections which were playable from a biomechanical standpoint but were unfavorable to performers. The opposite case appeared as well with the expert sometimes preferring difficult fingerings over easier ones. Radicioni et al. explain this phenomenon by the omitted modeling of the cognitive aspects of guitar play in their approach.

While the concept behind the presented cost function appears promising, its implementation features many parameters. This is negligible if a sufficient amount of training data is available or if extensive parameter search is conducted to find a reliable parameter set. However, Radicioni et al. neither apply a well-founded learning strategy nor tell details about their reference data used for experimentation. Additionally, the size of the test set is relatively low. As a consequence, the results of the presented cost function could vary in quality depending on the parameter set or the pieces for which fingerings are generated.

Modeling advanced polyphony

Combining their previous work, Radicioni et al. [19] present a path-based approach to the guitar fingering problem with the intent of supporting any given guitar piece. In addition to supporting melodies and chords they aim to support melodies accompanied by chords as well.

The first step of their approach is to automatically categorize passages of a piece into one of the categories `MEL`, `CHO` or `MIX` (melody, chord or a mixture of both). `MEL` and `CHO` passages are defined as a sequence of non-overlapping notes and respectively as a set of held notes starting and stopping together. A `MIX` passage needs to conform to both `MEL` and `CHO` with the added limitation that the melody must lie temporally within the start and end of the held notes. This limitation keeps the approach from supporting *any* piece because chords with notes starting or ending at different times (Radicioni et al. call these "polyphonic textures") are left out.

Their second step is to generate a layered graph from the categorized passages. `MEL` and `CHO` are dealt with as described in their previous work ([20] and [22]). When playing a `MIX` passage, the strings of the held notes need to be kept depressed while the melody notes are played. Therefore, a preliminary step to handling `MIX` passages is to generate all fingerings of the held notes in the same way it is done for chords. Each one of the held note fingerings is then used as a prerequisite constraint for generating matching melody fingerings. The fingerings of the held notes and each of their melodies are added to the layered graph as subgraphs.

Their third step is to generate the optimal fingering using the cost function described in [20] and a dynamic programming algorithm for inference on the layered graph. After describing their approach, they prove that the algorithmic complexity of this approach lies in $\mathcal{O}(n)$ given a piece of n events (an event being the start or end of a note).

The approach was evaluated similarly to their previous paper covering chords [21], meaning a comparison between generated fingerings and fingerings from a guitar expert is run. Their test set consists of excerpts of six guitar pieces of which three were already tested in [21]. For each kind of passage, two representative test pieces were chosen. An accuracy of 86.18 % was achieved on the individual fingered positions; less than in their previous paper. Radicioni et al. trace this deterioration in quality back to the performance achieved on `MIX` passages for which their system performs worse than for `MEL` and `CHO` passages. The overall accuracy reaches 97.50 % when finger assignments are excluded from comparisons, i. e. when comparing only fretboard positions. Radicioni et al. conclude that their strategy for the guitar fingering problem is appropriate based on these numbers. Future work may incorporate tempo information and physical dimensions of a performer and the instrument to correct shortcomings in the cost function.

Radicioni et al. succeed in modeling a common pattern to many guitar pieces for path-based approaches, widening their scope of applicability. The only passages left unsupported are polyphonic textures.

The number of test pieces is still relatively low compared to what one would expect for an evaluation of a complex system. Because the work of this paper depends on the cost function described in [21], the same comments apply.

Incorporating tempo information in the cost function

Hori et al. [9] reformulate the guitar fingering problem for the use with hidden markov models (HMM) and introduce a cost function which includes the duration between notes. Their system is catered to beginners at guitar playing.

A HMM is a probabilistic graphical model defined by a set of output symbols, a set of states (called hidden states), the transition probabilities between them, their initial probabilities and for each state and each symbol a probability how likely it is that a certain state emits a certain output symbol [18]. The decoding problem (the inference mechanism for HMMs) involves finding the sequence of hidden states which is most likely to emit a given sequence of observed output symbols. A transition between two states depends only on those two states without influence of states or symbols encountered earlier, meaning the Viterbi algorithm can be used for the inference task. HMMs are usually trained using known sequences of hidden states and output symbols. More details on the matter can be found in [3, 12, 18].

Hori et al. [9] restrict their approach to guitar pieces which can be expressed as a sequence of chords. Similar to the definition of `CHO` passages by Radicioni et al. [20], every note of a chord must start and end at the same time. Hori et al. coin the term *form* for all fingered positions simultaneously played by the left hand. The guitar fingering problem is mapped to HMMs by associating forms with hidden states and chords with output symbols. This way, finding the most likely sequence of hidden states results in the most likely sequence of forms which emit the chords of a piece. The transition probabilities between states encode the difficulty of transitioning between forms and the difficulty of the destination form. Which form creates which chord is established through the emission probabilities of the states. There is no mention of how the forms themselves are generated or if a layered graph is used internally to reduce the problem size. It is stated however that the HMM "has a huge number of hidden states" [9, p. 3] which leads one to assume that no simplifications were implemented.

In line with the work of other authors, Hori et al. define the state transition probabilities manually. Hand movement along the fretboard is modeled by a Laplace distribution whose variance is proportional to the time interval available for the movement. This term is multiplied by three other terms which represent the difficulty of the destination form, namely its fretwise span, the number of fingers involved in the form and the fret in which the index finger resides (to penalize playing in higher frets).

A qualitative comparison was conducted between the results of the HMM approach and two instances of tablature editing software with capabilities of generating fingerings. The HMM approach fared better than the competitors on two scales and a short polyphonic piece by using open strings more efficiently and generally requiring less movement of the left hand. The consistency of the fingerings to published sheet music was not analyzed.

The approach presented by Hori et al. differs from previous path-based approaches in several areas. Instead of employing a custom graph or inference technique Hori et al. resort to HMMs. They refrain from modeling transitions of individual fingers and instead resort to characterizing basic attributes of forms. This decision results in a concise definition of the state transition probabilities free from assumptions and experimentally determined constants. Unfortunately, the results of Hori et al. are incomparable to the results of other authors due to different evaluation techniques.

3.2 Evolutionary Approaches

Solutions to the guitar fingering problem based on genetic algorithms were presented by Rutherford [24], Tuohy et al. [31], and Tuohy et al. [33]. Their works are summarized in abridged form with a focus on their results and relevant aspects to path-based approaches.

Genetic algorithms are a technique for finding optimal solutions according to a non-convex cost function in a large search space. Optimization starts with a random population, i. e. a set of valid but unoptimized solutions. One evolutionary step involves calculating the quality of every solution by means of a "fitness function" and afterwards recombining the solutions among each other. An optimization across the population is achieved by using better solutions more often in the recombination process. The evolutionary step is repeated for a fixed number of iterations or until the improvements after each step stagnate. See [4] for more details on evolutionary algorithms.

Compared to path-based approaches, genetic algorithms have the advantage of optimizing solutions as a whole instead of optimizing smaller parts. This provides a greater flexibility for the definition of the cost function (resp. the fitness function). Disadvantages include the high number of iterations necessary until the algorithm converges to good solutions [24]. Many parameters such as the size of the population, the number of iterations and the recombination strategy are problem-dependent and need to be optimized separately.

The basic concept

Tuohy et al. [33] are the first to apply genetic algorithms (GA) to the guitar fingering problem. The mapping of the guitar fingering problem to GAs is straight-forward with several fingerings constituting a population. Contrary to path-based approaches, no independence assumptions between neighboring nodes need to be established. Tuohy et al. [33] examine only a subproblem of the guitar fingering problem, namely the assignment of fretboard positions to notes – fingers are abstracted from in their approach. The fitness function is influenced by the work of Heijink et al. [7] and consists of factors characterizing the difficulties of hand movement and hand manipulation. Included factors are the number of notes played as open strings, the fret distance between neighboring notes/chords and a factor penalizing index finger barre chords. It is not stated how barre chords were penalized without modeling fingers. Detailed information is given on the implementation of the recombination strategy and the manually chosen parameters for the GA.

Tuohy et al. discuss the quality of their results for 20 guitar pieces in comparison to published tablature and the generated tablature of a commercial tablature editor. Whereas the commercial software regularly generates uncomfortable or unplayable tablature, their approach mostly generated tablature close to the published reference tablature. For this reason, Tuohy et al. consider their presented approach to be successful.

Fitness function improvements and support for finger assignments

Tuohy et al. [31] refine their work by employing a distributed genetic algorithm (DGA) and by learning the weights of the fitness function by meta optimization. They also present a separate approach for assigning fingers to fretboard positions.

Each piece processed by their GA is manually segmented into "logical phrases" which are assumed to have no influence on each other. Following the elaboration of their DGA, baseline tabulatures are established by running a hand-tuned DGA on "eleven different musical excerpts" [31, p. 5]. The fitness function is comprised of 14 individually weighted factors for biomechanical and cognitive aspects of guitar play, again without explicitly modeling fingers or distances between them. The weights of the fitness function were meta optimized by a GA with reference to excerpts of 30 guitar pieces with known fretboard positions. The parameters of the DGA used for finding fretboard positions were also meta optimized on 11 excerpts. The composition of the datasets is not specified. Fingers are assigned to fretboard positions in a separate process by an artificial neural network (ANN) with 59 input variables and 4 output variables.

ANNs are a machine learning technique inspired by the biological neural networks found in nature [3]. An ANN consists of neurons which receive binary or real-valued messages from other neurons as an input. The sum of those messages is inserted into an *activation function* which determines the value of the output message sent to other neurons. Every input received from another neuron is associated with a weighting factor. The most common architecture of ANNs are multilayer perceptrons which feature one layer of input neurons, one layer of output neurons and a varying amount of hidden layers in between. The neurons are connected to each other layer by layer. As soon as input values are supplied to the neurons of the input layer, messages propagate through the ANN until neurons of the output layer return their messages. ANNs are trained by adjusting the weights of the neuron connections to training data for which the values of input and output neurons are known. More information on ANNs can be found in Bishop [3].

The input variables are explained in detail. They cover the fretboard position of a single note and information on the positions of surrounding notes. The four output variables represent the four fingers of the left hand used for playing the guitar. Finding the finger best suited for a note is achieved by feeding the input values of the note into the network and observing the output variable with the highest value. The ANN was trained and tested on 30 pieces with fingerings taken from classtab.org[1)], totalling 6800 notes. Tuohy et al. employ another meta GA to select the most significant of the 59 input variables (amongst other ANN parameters). 70 % of the notes in the dataset went into training the ANN while 15 % served as a reference for assessing the fitness of the ANN. The remaining 15 % constitute the test set. No information is given on the outcome of this optimization.

Tuohy et al. analyze the accuracy of their DGA for fretboard positions by counting the number of times the previously established baseline solution was found within ten runs. The DGA outperforms their previously published GA with 74.7 % accuracy compared to 40 % accuracy.

The DGA achieved an accuracy of 91.1 % "on [their] test set" [31, p. 9] when comparing fretboard positions for consistency. Tuohy et al. report no significant increase in tablature difficulty for mismatching fretboard positions. The accuracy of the ANN for assigning fingers to fretboard positions was 80.6 % on the 15 % of the dataset put aside earlier. Tuohy et al. comment on the 90.61 % accuracy achieved by Radicioni et al. [21] stating that the chosen guitar pieces are easy cases for generating tablature. Their

1) http://www.classtab.org, a moderated library of classical guitar tablature anyone can contribute to

DGA scored 98.9 % accuracy on the same pieces which Tuohy et al. regard as a confirmation of the quality of their approach.

Tuohy et al. [31] are the first to include a direct comparison to the results of another author, establishing a ranking in the otherwise incomparable results of previous literature. Valuable information was omitted in some areas, such as the optimized selection of input variables for the ANN or the composition of the various datasets cited throughout. It remains unclear which dataset was used for the final evaluation of the DGA and why the DGA and the ANN were evaluated on different datasets. A detailed description of the data would have benefited the comparability of the approach, especially since its results are so encouraging by the numbers.

3.3 Neural-Based Approaches

Tuohy et al. [32] present another approach using neural networks, where the network generates fretboard positions instead of assigning fingers.

Their network features 64 inputs and 20 outputs which are covered in detail. The information fed into the network is based on a single note, on preceding notes for which the fretboard position was already determined and on characteristics of succeeding notes. Each network output corresponds to a fret on the guitar. Once the relevant information on a note is fed into the network, an approximate fret position can be read from the network outputs. Tuohy et al. then determine an exact fretboard position by finding the nearest position to the approximate fret creating the note.

The training data for the network initially consisted of 75 excerpts of guitar tablature taken from classtab.org. As a measure to filter out noisy training data and irrelevant input states, Tuohy et al. decided to employ meta optimization to select a subset of 65 excerpts and the subset of the input states which, when used for training an ANN, result in the best accuracy on an independent test set. As in their previous work, genetic algorithms are used for this purpose. Neither the source and composition of the independent test set nor the selected network inputs are reported. However, the 65 excerpts were uploaded to the author's website.

Because the ANN occasionally misplaces notes Tuohy et al. introduce a second pass over the generated tablature to correct obvious mistakes. The correction pass repurposes their GA fitness function from [31].

Tuohy et al. evaluate their ANN (trained on the 65 most significant excerpts) on the same 65 excerpts, reaching an accuracy of 91.4 % when comparing fretboard positions for consistency. The accuracy after applying the correction pass is 91.1 %. Their GA approach from [31] achieves 86.9 % accuracy on the same data. The runtime of their algorithms is stated briefly. Generating tablature for a short excerpt (which is included in the paper) takes 5.7 s with their GA and 55.6 ms with the ANN. No information is included on the system used to obtain these measurements.

Since their GA and ANN approaches rely only on tablature, Tuohy et al. expect an improvement in accuracy once tempo and note duration are included in the calculations.

Tuohy et al. present an ANN solution to (a simplified version of) the guitar fingering problem with a reportedly high accuracy. It must be noted however, that the ANN was tested on the training data which severely affects the significance of the presented results. Tuohy et al. are the first to provide the full dataset used for their research. The accuracy of their GA on this dataset was used as a baseline in this thesis.

3.4 Summary

Many different algorithms have been applied to the guitar fingering problem over the years, each with their own advantages and disadvantages. Path-based approaches are efficient at finding fingerings but suffer from the locality of their cost function introduced by the necessary independence assumption between notes. Evolutionary approaches remedy the cost function weakness but require many iterations even for short pieces. Artificial neural networks were applied, solving either only a subproblem or generating fingerings of uncertain quality.

One can hardly single out an approach as being the best among the presented work due to the heterogenous problem formulations, data sources and evaluation techniques employed. By the numbers, the works of Tuohy et al. [31] create the best fingerings, but on an unknown test set. No extensive evaluation with more than three guitar experts has yet been conducted to judge the artistic quality of generated fingerings.

Most of the difficulty aspects described in section 2.5 were addressed, albeit in an abstract sense: Until now, distances on the guitar were always modeled as fret distances instead of true metric distances and force was only implicitly represented by difficulty factors instead of assigning difficulty to exerting a certain amount of force. Knowledge on tempo and note durations was so far only incorporated by Hori et al. [9].

In conclusion, the guitar fingering problem cannot be considered as being entirely researched yet and offers room for investigations or improvements in several areas.

3.5 Thesis Goals

The goal of this thesis is to expand the capabilities of path-based approaches to solve the guitar fingering problem by

- lifting the restriction of path-based approaches only being applicable to specifically structured guitar pieces (such as melodies or chord sequences) and by
- experimenting with including the physical dimensions of guitars and guitarists in the cost function.

Furthermore, the approach should take advantage of note values and the tempo of a piece.

Linear-chain conditional random fields were chosen as the underlying model because

- they offer *features* as modular cost function components whose individual weight can be optimized to fit labeled training data, and because
- the cost function components have access to more knowledge than is available for the definition of the state transition probabilities in HMMs.

Throughout this thesis, a focus is laid on reproducibility and attention to details, since from the personal opinion of the author, these aspects are neglected in other publications.

4 Linear-Chain Conditional Random Fields

This chapter serves as a short summary of linear-chain conditional random fields based on the introduction on the topic written by Klinger et al. [12]. A full explanation can be found in their paper.

Notation

Sets: Sets are written in all capital letters. The size of a set Y is expressed as $|Y|$. The empty set is written as $\emptyset$.

Vectors: Unless specified otherwise, a vector consisting of elements $y \in Y$ is written as $\vec{y}$. Access to the i-th item of a vector is written as $\vec{y}_i$.

Named tuples: Given a set of tuples $T = \{ (a, b) \mid a \in A, b \in B \}$ for any given sets A, B, access to the a-component of a tuple $t \in T$ is written as $\mathrm{a}(t)$.

Variable names: Variable names are chosen to be consistent with published literature where possible, especially for the topic of conditional random fields.

4.1 Definition

Conditional random fields (CRFs) are a type of probabilistic graphical model initially described by Lafferty et al. [13].

Probabilistic graphical models consist of a set of nodes and edges, where nodes represent random variables whose dependencies to other variables are indicated by edges between the nodes [12, p. 10]. Two nodes not connected by an edge represent conditional independence between the random variables [12, p. 10]. Two random variables a and b are conditionally independent iff $p(a,b|c) = p(a|c) \cdot p(b|c)$ for any other random variable c [12, p. 10]. Consequently, the layout of the graph represents independence properties of the probability distribution defined by its random variables, hence the name *independency graph* [12, pp. 10–11].

CRFs model the probability $p(\vec{y}|\vec{x})$ for several labels (or outputs) $\vec{y} = (y_1, y_2, \ldots, y_n) \in \mathfrak{Y}^n$ and corresponding observations (or inputs) $\vec{x} = (x_1, x_2, \ldots, x_n) \in \mathfrak{X}^n$ where $\mathfrak{Y}$ is called *label alphabet* and $\mathfrak{X}$ is the set of observations [12, p. 14]. In the case of linear-chain CRFs, the output variables are arranged as a linear chain in the independency graph (see figure 6), meaning the output variables represent a sequence [12, p. 15].

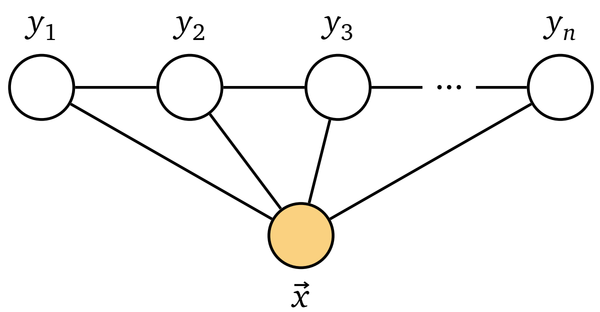


Figure 6.: Independency graph of a linear-chain CRF – the variables $\vec{x}$ are shaded because their values are known (adapted from Klinger et al. [12, figure 6 (a)])

The independency graph conveys the fact the linear-chain CRFs adhere to the first-order Markov assumption: Any output variable y_t depends only on the immediate predecessor y_{t-1} and $\vec{x}$ but is independent from the output variables $y_{t-2}, y_{t-3}, \ldots, y_1$.

Linear-chain CRFs can be used to find the most likely label sequence $\vec{y}^*$ to a given observation sequence $\vec{x}$ [12, p. 19]. More formally, this corresponds to finding $\vec{y}^* = \text{argmax}_{\vec{y} \in \mathcal{Y}}\, p(\vec{y}|\vec{x}, \mathcal{M})$ for a given linear-chain CRF $\mathcal{M}$, an observation sequence $\vec{x}$ and the set of all possible label sequences $\mathcal{Y}$.

The probability of $\vec{y}$ given $\vec{x}$ with $|\vec{y}| = |\vec{x}| = n + 1$ is defined as [12, Eq. 39, 44, 42]:

$$\begin{aligned} p_{\vec{\lambda}}(\vec{y}|\vec{x}) &= \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \cdot \prod_{j=1}^{n} \Psi_j(\vec{x}, \vec{y}) \\ &= \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \cdot \prod_{j=1}^{n} \exp\left(\sum_{i=1}^{m} \lambda_i f_i\big(y_{j-1}, y_j, \vec{x}, j\big) \right) \\ &= \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \cdot \exp\left(\sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i\big(y_{j-1}, y_j, \vec{x}, j\big) \right) \end{aligned}$$

$Z_{\vec{\lambda}}(\vec{x})$ normalizes the term by summing over $\mathcal{Y}$ and is defined as [12, Eq. 40, 43]:

$$Z_{\vec{\lambda}}(\vec{x}) = \sum_{\vec{y} \in \mathcal{Y}} \prod_{j=1}^{n} \Psi_j(\vec{x}, \vec{y}) = \sum_{\vec{y} \in \mathcal{Y}} \exp\left(\sum_{j=1}^{n} \sum_{i=1}^{m} \lambda_i f_i\big(y_{j-1}, y_j, \vec{x}, j\big) \right)$$

$f_i \in \mathcal{F}$ are problem-specific *feature functions* which establish the connection between labels and observations. Arguments to a feature function are two subsequent labels y_{j-1}, y_j from $\vec{y}$, the entire observation sequence $\vec{x}$ and the current position within the sequence j. This stands in contrast to HMMs which have only access to y_j, y_{j-1} and $\vec{x}_j$ to define the state transition probabilities. A CRF of higher order would provide the feature functions with a longer history of labels [12, p. 19]. Feature functions need not to conform to a specific co-domain because the enclosing exponentiation ensures the positivity of the whole term [12, p. 13].

Each feature function f_i is weighted by an individual factor λ_i. Finding the optimal weighting $\vec{\lambda}^*$ is the main training task of CRFs [12, p. 19].

$\mathcal{Y}$ depends on the problem and can be any subset of $\mathfrak{Y}^n$.

A linear-chain CRF can be represented by a stochastic finite state automaton (SFSA, alternatively called probabilistic automaton [15]) [12, p. 18]. An SFSA is comprised of a set of states S, a subset of initial states and directed edges $T \subseteq S \times S$. Edges are associated with a transition probability $p_{s,s'}(\sigma)$ depending on the two states $s, s' \in S$ connected and an additional input value $\sigma \in \Sigma$[15, p. 5][12, p. 18].

For linear-chain CRFs, each state $s_y \in S$ of the automaton represents a label y of the label alphabet $\mathfrak{Y}$ [12, p. 18]. The set of initial states can be chosen freely depending on the problem. An edge $s \rightarrow s'$ specifies that the label y' may follow y. For the transition probabilities, $\vec{x}$ takes the role of σ, resulting in:

$$p_{s,s'}(\sigma) = \frac{\Psi_j(\vec{x}, s, s')}{\sum_{s'' \in S} \Psi_j(\vec{x}, s, s'')} = \frac{\exp\left(\sum_{i=1}^{m} \lambda_i f_i(y, y', \vec{x}, j)\right)}{\sum_{y'' \in \mathfrak{Y}} \exp\left(\sum_{i=1}^{m} \lambda_i f_i(y, y'', \vec{x}, j)\right)}$$

4.2 Inference

The most likely output sequence $\vec{y}^* = \text{argmax}_{\vec{y} \in \mathcal{Y}}\, p(\vec{y}|\vec{x}, \mathcal{M})$ for a given linear-chain CRF $\mathcal{M}$, an observation sequence $\vec{x}$ and the set of all possible label sequences $\mathcal{Y}$ can be computed efficiently with the Viterbi algorithm [12, p. 23] commonly used for HMMs [18].

Instead of finding $\vec{y}^*$ by exhaustive search over all $y \in \mathcal{Y}$, the Viterbi algorithm employs dynamic programming, i. e. it reuses partial solutions to reduce the number of computation steps necessary to find a solution.

An observation sequence $\vec{x}$ of n elements requires n iterations. In each iteration $j \in [1,n]$ the maximum achievable score $\delta_j(s')$ of a path through the SFSA ending at $s' \in S$ is computed. Because transitions $s \rightarrow s'$ in SFSAs are independent of the history of previous states, $\delta_j(s')$ can be computed as the maximum over $s \in S$ of the score $\delta_{j-1}(s)$ of the immediate predecessor state s times the score $\Psi_j(\vec{x}, s, s')$. Another set of variables $\psi_j(s')$ keeps track of the respective predecessor state s which is responsible for the maximum score stored in $\delta_j(s')$. Once the algorithm has reached $\delta_n(s')$ the most likely state sequence can be retrieved by following the best predecessors stored in $\psi_j(s')$.

Technically, the values $\delta_j(s')$ are not probabilities because the results of $\Psi_j(\vec{x}, s, s')$ are not normalized to $[0,1]$. It is legitimate to omit the normalization because it has no influence on finding the most likely output sequence.

The algorithmic complexity of the Viterbi algorithm is $\mathcal{O}(|S|^2 \cdot n)$ [12, p. 23].

Given that all labels $y \in \mathfrak{Y}$ may follow each other and given S as the set of start states, the exact steps of the algorithm are [12, p. 24][18, p. 8]:

1. Initialization with the dummy start state $\perp$:

$$\forall s' \in S: \quad \delta_1(s') = \Psi_1(\vec{x}, \perp, s') \\ \psi_1(s') = \perp$$

2. Recursion for $j = 2, \ldots, n$:

$$\forall s' \in S: \quad \delta_j(s') = \max_{s \in S} \delta_{j-1}(s) \cdot \Psi_j(\vec{x}, s, s') \\ \psi_j(s') = \operatorname*{argmax}_{s \in S} \delta_{j-1}(s) \cdot \Psi_j(\vec{x}, s, s')$$

3. Termination:

$$\vec{y}_n^* = \operatorname*{argmax}_{s \in S} \delta_n(s)$$

4. State sequence backtracking for $j = n-1, n-2, \ldots, 1$:

$$\vec{y}_j^* = \psi_{j+1}(\vec{y}_{j+1}^*)$$

Its mode of operation can be shown in a trellis diagram (or lattice diagram [3]) as depicted in figure 7.

4.3 Training

The optimal feature weights $\vec{\lambda}$ of a linear-chain CRF are determined by maximum likelihood estimation. The log-likelihood $\mathcal{L}$ is maximized on labeled observation sequences $\mathcal{T}$ with respect to $\vec{\lambda}$ [12, p. 19]:

$$\mathcal{L}(\mathcal{T}) = \sum_{(\vec{x},\vec{y}) \in \mathcal{T}} \log p(\vec{y}|\vec{x})$$

Overfitting[1)] is avoided by regularization, i. e. by adding a penalty term to $\mathcal{L}(\mathcal{T})$ [3, p. 10]. The term used for linear-chain CRFs is [12, pp. 19–20]

$$-\sum_{i=1}^{m} \frac{\lambda_i^2}{2\sigma^2}$$

1) A model adjusting so well to the training data that its performance on unseen data suffers [35, p. 145].

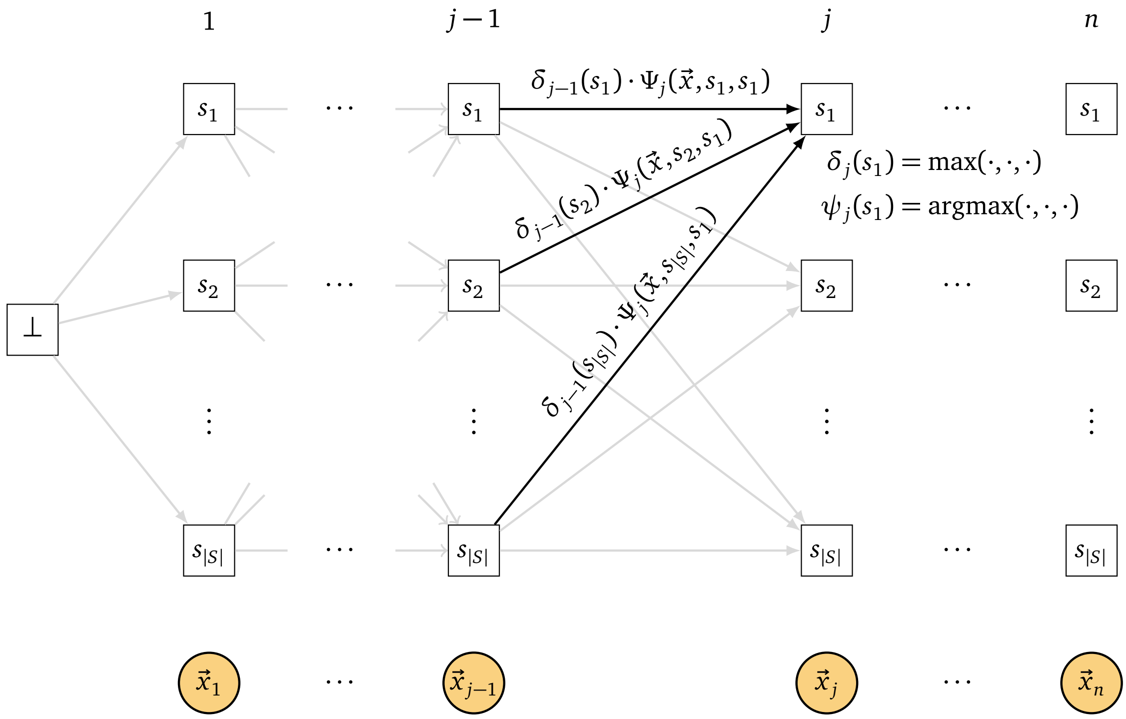


Figure 7.: Trellis diagram visualizing the recursion step of the Viterbi algorithm (adapted from Klinger et al. [12, figure 9])

and prevents features from dominating the calculation by penalizing high weights. The influence of the penalty term increases the lower σ^2 is chosen [12, p. 20]. Note that feature functions of homogenous co-domain are a prerequisite for this penalty term to operate correctly because otherwise, a high value λ_i would not necessarily imply a high impact of the feature f_i.

In the following, only the most important aspects regarding the training of a CRF are outlined, refer to Klinger et al. [12] for details.

The derivation of $\mathcal{L}$ is elaborated in section 4.2.1. of Klinger et al. [12]. $\frac{\partial \mathcal{L}}{\partial \vec{\lambda}}$ (including the penalty term) exhibits only one extremum because it is a concave function. This extremum can be found by numerical optimization.

Evaluating $\frac{\partial \mathcal{L}}{\partial \vec{\lambda}}$ during the optimization requires an application of the forward-backward algorithm for each tuple $(\vec{y}, \vec{x}) \in \mathcal{T}$ [12, p. 22]. The forward-backward algorithm is a dynamic programming algorithm similar to the Viterbi algorithm used for computing $p(y|\vec{x})$ for all $y \in \mathfrak{Y}$ [18]. It relies on the same SFSA and trellis [18]. Analogous to the Viterbi algorithm, its algorithmic complexity is $\mathcal{O}(|S|^2 \cdot n)$.

5 Application of Linear-Chain CRFs on the Guitar Fingering Problem

The problem statement (recapitulated from section 1.1) is to assign a string, fret and finger to each note of a given guitar piece to obtain an optimal fingering according to a quality measure. A rough description of the application of CRFs for this purpose is:

- Given a guitar piece, choose the sequence of notes as the observation sequence supplied to the CRF.
- Encode string, fret and finger as the labels of the CRF.
- Provide feature functions which encode the difficulty aspects of guitar playing.
- Compute the most likely label sequence to obtain the optimal fingering of the piece.

The sections of this chapter cover the implementation of this principle in greater detail: Section 5.1 specifies elements of the guitar domain for the use with CRFs. The application of CRFs is first shown for guitar pieces of one voice in section 5.2. The subsequent section demonstrates the application on pieces of multiple voices.

5.1 Formal Specification of the Guitar Domain

5.1.1 Basic Definitions

The smallest pitch increment notated in classical music is one semitone. Define the set of pitches $\mathbb{P}$ as $\mathbb{P} = \mathbb{Z}$ so that a pitch $p \in \mathbb{P}$ is represented by a number of semitones. The mapping between a pitch and a value in $\mathbb{Z}$ can be chosen arbitrarily: In this thesis, let 0 represent the pitch C0. To improve readability, pitches are referred to by their names instead of their numerical value in $\mathbb{P}$.

The maximum number of pitches in a chord is restricted by the number of strings of a classical guitar (6 strings). Define a chord as a vector of notes $\vec{p} = (p_1, \ldots, p_c) \in \mathbb{P}^c$ for $c \in [1,6]$. Let $tu = (p_1, \ldots, p_6) \in \mathbb{P}^6$ denote a tuning for a classical guitar where p_i is the pitch of the i-th string.

Because classical guitars have 19 frets, a position on the fretboard is defined by a string $s \in [1,6]$ and a fret $fr \in [0,19]$. $fr = 0$ denotes an open string analogous to guitar tablature. Let $fi \in [0,4]$ denote a finger of the left hand with same semantic as in sheet music and tablature, meaning the index finger is identified by $fi = 1$, the middle finger by $fi = 2$ etc. and $fi = 0$ is the finger reserved for playing open strings. A *fingered position* encodes a finger of the left hand depressing a certain position on the fretboard and is defined as a 3-tuple $\langle s, fr, fi \rangle$ of a string s, a fret fr and a finger fi. This notation matches the notation chosen by Radicioni et al. [19]. The set of fingered positions is defined as:

$$POSITIONS = \left\{ \langle s, fr, fi \rangle \;\middle|\; s \in [1,6], fr \in [0,19], fi \in [0,4] \land fr = 0 \Leftrightarrow fi = 0 \right\}$$

Analogous to Hori et al. [9], let a *form* denote the fingering of a single chord defined by a vector of fingered positions $\vec{pos}$. A form is *consistent* to a chord $\vec{p}$ with respect to a tuning tu if the i-th fingered position produces the i-th pitch of the chord:

$$\text{consistent}(\vec{pos}, \vec{p}, tu) \Leftrightarrow |\vec{pos}| = |\vec{p}| \land \forall i \in \left[1, |\vec{p}|\right] : tu_{s(\vec{pos}_i)} + \text{fr}(\vec{pos}_i) = \vec{p}_i$$

To conclude: A consistent form $\vec{pos}$ establishes a 1:1 mapping of fingered positions to the pitches of a chord $\vec{p}$ where the order of the positions follows the order of the pitches inside the chord.

5.1.2 Definition of Playability

More predicates are necessary to narrow down the subset of forms which are playable for guitarists from a biomechanical standpoint.

As mentioned in section 3.1, Radicioni et al. [19] already presented an approach for generating playable forms by the means of a constraint satisfaction problem (CSP). Their CSP supports index finger barre chords and forms which require stretching of the left hand to reach the positions on the fretboard. Radicioni et al. [19] describe the constraints in textual form. Both the constraints and the CSP approach were adapted in this thesis. The following paragraphs explain these constraints in detail and show a formal definition of how they were implemented.

1. One note per string

A form whose fingered positions are consistent to a chord can only produce the pitches of the chord if all its positions are located on different strings (obviously only one pitch can be played on one string at a time) [19]. Given a form $\vec{o}$, $|\vec{o}| = c$, this can be enforced by:

$$\text{one\_note\_per\_string}(\vec{o}) \iff \forall i,j \in [1,c], i \neq j : \text{s}(\vec{o}_i) \neq \text{s}(\vec{o}_j)$$

2. Comfortable finger order

When assigning fingers to positions on the fretboard, the anatomical order of the fingers should be respected so that the fingers do not cross. A finger identified by a higher number than another finger should be placed in the same fret or in a higher fret [19]. If two fingers are located in the same fret, the higher finger should be placed on a lower numbered string, i. e. behind the lower finger when seen from the point of view of the guitarist. This rule can be explained by the the fact that guitarists do not hold their left hand exactly parallel to the fretboard. Instead, the hand is rotated slightly towards the corpus of the guitar so that the index finger is oriented at a 45° angle relative to the strings [7]. Placing higher fingers on higher numbered strings in this posture would otherwise lead to crossed fingers.

Given a form $\vec{o}$ with $|\vec{o}| = c$, this principle can be modeled by:

$$\begin{aligned}\text{finger\_order}(\vec{o}) \iff \forall i,j \in [1,c], i \neq j : &\Big(\text{fr}(\vec{o}_i) < \text{fr}(\vec{o}_j) \Longrightarrow \text{fi}(\vec{o}_i) < \text{fi}(\vec{o}_j)\Big) \land \\ &\Big(\text{fr}(\vec{o}_i) = \text{fr}(\vec{o}_j) \Longrightarrow \big(\text{s}(\vec{o}_i) < \text{s}(\vec{o}_j) \land \text{fi}(\vec{o}_i) \geq \text{fi}(\vec{o}_j)\big) \lor \\ &\qquad \big(\text{s}(\vec{o}_i) > \text{s}(\vec{o}_j) \land \text{fi}(\vec{o}_i) \leq \text{fi}(\vec{o}_j)\big)\Big)\end{aligned}$$

3. Distances between fingers

Many biomechanially unplayable forms can be ruled out by restricting the total fret span of a form. This measure is not sufficient on its own because forms of a relatively low fret span can nonetheless include unplayable distances between certain finger pairs. The distance of how far two fingers can be spread apart varies between the possible finger pairs [34]. The distance is especially low for the pair of middle and ring finger [34]. Although it would be possible to model the finger and fretboard distances in metric units, modeling distances by an integer of frets is computationally more favorable for constraint satisfaction problems.

Determining the maximum fret spans for each finger pair in the first fret would result in relatively low values due to the large gaps between frets. These values would falsely ban forms played in higher frets where the gaps between frets are smaller. As a consequence, the fret spans need to be determined near or on the corpus. The maximum fret spans implemented in this thesis are based on the values reported by Radicioni et al. [19] and are shown in table 1. One span was increased because it was found to be playable in higher frets.

	index (1)	middle (2)	ring (3)	little (4)
little (4)	4	3	1	–
ring (3)	3	**1**	–	
middle (2)	2	–		
index (1)	–			

Table 1.: Maximum permitted number of frets lying between finger pairs (see Radicioni et al. [19, table 1]) – Modifications to the spans used by Radicioni et al. are emphasized.

Given table 1 and a form $\vec{o}$ with $|\vec{o}| = c$, the predicate for the finger distances can be formulated as:

$$\text{finger\_distance}(\vec{o}) \iff \forall i,j \in [1,c], i < j : \text{fr}(\vec{o}_i) = 0 \lor \text{fr}(\vec{o}_j) = 0 \lor \left|\text{fr}(\vec{o}_i) - \text{fr}(\vec{o}_j)\right| \leq \text{max\_distance}\big(\text{fi}(\vec{o}_i), \text{fi}(\vec{o}_j)\big) + 1$$

Radicioni et al. [21] reportedly enforced minimum fret spans between finger pairs but do not go into detail on how they were implemented. The concept of minimum spans was omitted in this thesis because it was found to be hard to represent without a more sophisticated model of the left hand.

4. Barre chords

Radicioni et al. [19] describe the barre constraint as "all the positions of the barre are on the same fret and all the other positions in the chord are in higher-numbered frets" [19, Table 1]. This constraint is decomposed into individual predicates in the following.

Up to this point, no predicate has restricted the number of appearances of each finger yet. Since it should be possible to play forms with multiple open strings and index finger barre chords, the fingers 0 and 1 should be the only fingers allowed to appear multiple times in a form. Given a form $\vec{o}$ with $|\vec{o}| = c$, define:

$$\text{allow\_only\_index\_finger\_barre}(\vec{o}) \iff \forall fi \in \{2,3,4\} : \neg\exists i,j \in [1,c], i \neq j : \text{fi}(\vec{o}_i) = fi \land \text{fi}(\vec{o}_j) = fi$$

Requiring that all positions of a barre are located in the same fret is equal to banning diagonal barres (see section 2.4). Apart from the fact that diagonal barres are hard to model (the positions involved have to be arranged approximately in a line, the line depends on the dimensions of a performer's finger and the instrument dimensions, etc.), they are a rare occasion and are therefore negligible to omit. Given a form $\vec{o}$ with $|\vec{o}| = c$, restrict fingers to one fret by:

$$\text{no\_diagonal\_barre}(\vec{o}) \iff \forall i,j \in [1,c], i \neq j : \text{fi}(\vec{o}_i) = \text{fi}(\vec{o}_j) \implies \text{fr}(\vec{o}_i) = \text{fr}(\vec{o}_j)$$

The requirement of other positions being located in higher frets was split up in two predicates. Also, an auxiliary predicate was necessary which states whether a form contains an index finger barre:

$$\text{is\_index\_finger\_barre}(\vec{o}) \iff \exists i,j \in [1,c], i \neq j : \text{fi}(\vec{o}_i) = 1 \land \text{fi}(\vec{o}_j) = 1$$

The first predicate ensures that any position located on a lower numbered string than a position involved in a barre has to reside in the same fret or above:

$$\text{no\_lower\_frets}(\vec{o}) \iff \forall i,j \in [1,c], i \neq j : \big(\text{is\_index\_finger\_barre}(\vec{o}) \land \text{fi}(\vec{o}_i) = 1 \land \text{s}(\vec{o}_j) < \text{s}(\vec{o}_i)\big) \implies \text{fr}(\vec{o}_i) \leq \text{fr}(\vec{o}_j)$$

The second predicate bans partially depressed barres. The predicate is defined so that all fingers involved in a barre fret have to be played by the index finger:

$$\text{no\_partial\_barre}(\vec{o}) \iff \forall i,j \in [1,c], i \neq j : \Big(\text{is\_index\_finger\_barre}(\vec{o}) \land \text{fr}(\vec{o}_i) = \text{fr}(\vec{o}_j) \land \big(\text{fi}(\vec{o}_i) = 1 \lor \text{fi}(\vec{o}_j) = 1\big)\Big) \implies \text{fi}(\vec{o}_i) = \text{fi}(\vec{o}_j)$$

Combining all barre-related predicates into one yields:

$$\text{barre}(\vec{o}) \iff \text{allow\_only\_index\_finger\_barre}(\vec{o}) \wedge \text{no\_diagonal\_barre}(\vec{o}) \wedge \\ \text{no\_partial\_barre}(\vec{o}) \wedge \text{no\_lower\_frets}(\vec{o})$$

Complete predicate

A form $\vec{pos}$ is considered to be biomechanically playable if the following predicate holds:

$$\text{playable}(\vec{pos}) \iff \text{one\_note\_per\_string}(\vec{pos}) \wedge \text{finger\_order}(\vec{pos}) \wedge \text{finger\_distance}(\vec{pos}) \wedge \text{barre}(\vec{pos})$$

Note that a form $\vec{pos}$ includes only those fingered positions of a barre which contribute to the pitches of $\vec{p}$. Other positions on the fretboard may be depressed but are not explicitly included in the form.

Given the above predicate, one can specify the following sets:

$$PLAYABLE = \left\{ \vec{pos} \;\middle|\; \vec{pos} \in POSITIONS^c, c \in [1,6] \wedge \text{playable}(\vec{pos}) \right\}$$
$$PLAYABLE\_CONSISTENT_{\vec{p},tu} = \left\{ \vec{pos} \;\middle|\; \vec{pos} \in PLAYABLE \wedge \text{consistent}(\vec{pos}, \vec{p}, tu) \right\}$$

5.1.3 Generating Playable Forms

Obtaining the set $PLAYABLE\_CONSISTENT_{\vec{p},tu}$ for a given chord and tuning is necessary for the application of the linear-chain conditional random field covered in the next chapter. To this end, the CSP technique described by Radicioni et al. [19] was replicated:

Given a chord $\vec{p}$ and a tuning tu, define one triplet of variables $\langle s, fr, fi \rangle$ and their allowed domains for each pitch $\vec{p}_i$. Post the predicates stated above including consistent($\vec{pos}, \vec{p}, tu$) as the set of constraints of the CSP. Each variable assignment which satisfies all constraints represents a playable and consistent form $\vec{pos} \in PLAYABLE\_CONSISTENT_{\vec{p},tu}$. In case $\vec{p}$ is biomechanically unplayable, for example because it contains one low note and one high note whose only positions lie on different ends of the fretboard, the constraint satisfaction problem has no solution, meaning $PLAYABLE\_CONSISTENT_{\vec{p},tu} = \emptyset$.

A description of the search algorithms employed to find the solutions of CSPs is skipped here. Details on the matter can be found directly in the paper by Radicioni et al. [19] or more general in [29].

5.2 Application on Guitar Pieces of One Voice

This section demonstrates how CRFs can be applied to generate optimal fingerings for guitar pieces of one voice. Pieces of this type consist of a sequence of chords whose notes start and stop together, coinciding with the type of guitar piece supported by the HMM approach of Hori et al. [9]. Note values and tempo are abstracted from in this section and are introduced together with the support for guitar pieces of multiple voices in the next section.

CRF Definition

As described in section 4.1, CRFs are defined on a label alphabet $\mathfrak{Y}$ and observations $\mathfrak{X}$. In the guitar fingering problem, biomechanically playable forms should be assigned to chords, therefore

$$\mathfrak{Y} = PLAYABLE$$
$$\mathfrak{X} = \mathbb{P}^c \text{ for } c \in [1,6]$$

Given a piece $\vec{x} \in \mathfrak{X}^n$ as a sequence of chords, the possible label sequences are $\mathcal{Y} = \mathfrak{Y}^n$. The set of consistent fingerings $\mathcal{Y}_{\vec{x},tu} \subseteq \mathcal{Y}$ for $\vec{x}$ given a tuning $tu \in \mathbb{P}^6$ is defined as:

$$\mathcal{Y}_{\vec{x},tu} = \left\{ \vec{y} \mid \vec{y} \in \mathcal{Y} \wedge \forall j \in [1,n] : \text{consistent}(\vec{y}_j, \vec{x}_j, tu) \right\}$$

Note that $|\mathcal{Y}_{\vec{x},tu}| \ll |\mathcal{Y}|$ because $\mathcal{Y}$ consists of a vast amount of forms which create dissonant chords or chords of keys not appearing in $\vec{x}$ when played. Invalid fingerings $\vec{y}'$ with mismatching forms and chords originating from $\mathcal{Y}'_{\vec{x},tu} = \mathcal{Y} \setminus \mathcal{Y}_{\vec{x},tu}$ should be ruled out, meaning $p(\vec{y}'|\vec{x}) = 0$.

To this end, one can formulate a fully connected SFSA with one state $s_y \in S$ for each biomechanically playable form $y \in \mathfrak{Y}$. Analogous to the use of output probabilities in the HMM by Hori et al. [9], a CRF feature f_{cons} checks the consistency between forms and chords:

$$f_{\text{cons}}\Big(y_{j-1}, y_j, \vec{x}, j\Big) = \begin{cases} 0 & \text{if consistent}(y_j, \vec{x}_j, tu) \\ -\infty & \text{otherwise} \end{cases}$$

Assuming $\lambda_{\text{cons}} \in \mathbb{R}^+$ and given other features $f_i \in \mathcal{F}$ characterizing the difficulty of guitar playing and their weights $\vec{\lambda}$, this leads to:

$$\begin{aligned}
\forall \vec{y}' \in \mathcal{Y}'_{\vec{x},tu}: \qquad & p_{\vec{\lambda}}(\vec{y}'|\vec{x}) = \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \cdot \exp\left(\sum_{j=1}^{n}\left(\lambda_{\text{cons}} f_{\text{cons}}\Big(y_{j-1}, y_j, \vec{x}, j\Big) + \sum_{i=1}^{m} \lambda_i f_i\Big(y_{j-1}, y_j, \vec{x}, j\Big)\right)\right) \\
\Longleftrightarrow \qquad & = \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \cdot \exp(-\infty) \\
\Longleftrightarrow \qquad & = 0
\end{aligned}$$

for

$$Z_{\vec{\lambda}}(\vec{x}) = \sum_{\vec{y} \in \mathcal{Y}} \exp\left(\sum_{j=1}^{n}\left(\lambda_{\text{cons}} f_{\text{cons}}\Big(y_{j-1}, y_j, \vec{x}, j\Big) + \sum_{i=1}^{m} \lambda_i f_i\Big(y_{j-1}, y_j, \vec{x}, j\Big)\right)\right)$$

The probability of consistent solutions from $\mathcal{Y}_{\vec{x},tu}$ is not affected.

While this approach works in theory it is not feasible in practice: Computing the number of elements in $\mathfrak{Y}$ by counting the number of solutions to the constraint satisfaction problem specifying the elements of *PLAYABLE* yields over 250 000 000 solutions. Considering that the algorithmic complexity of both the forward-backward algorithm for training and the Viterbi algorithm for inference is quadratic in the number of states, it becomes apparent that working with an SFSA of over 250 000 000 states is not feasible.

Notice however that fingerings from $\mathcal{Y}'_{\vec{x},tu}$ do not contribute to the normalization $Z_{\vec{\lambda}}$ because their exponential term is zero, which means normalizing over $\mathcal{Y}_{\vec{x},tu}$ is equal to normalizing over $\mathcal{Y}$. The optimal fingering $\vec{y}^*$ for a piece $\vec{x}$ must be among $\mathcal{Y}_{\vec{x},tu}$ and no fingerings other than the ones found in $\mathcal{Y}_{\vec{x},tu}$ influence $p(\vec{y}|\vec{x})$. Therefore:

$$\begin{aligned}
\forall \vec{y} \in \mathcal{Y}_{\vec{x},tu}: \quad p_{\vec{\lambda}}(\vec{y}|\vec{x}) &= \frac{1}{Z_{\vec{\lambda}}(\vec{x})} \cdot \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\Big(y_{j-1}, y_j, \vec{x}, j\Big)\right) \\
Z_{\vec{\lambda}}(\vec{x}) &= \sum_{\vec{y} \in \mathcal{Y}_{\vec{x},tu}} \exp\left(\sum_{j=1}^{n}\sum_{i=1}^{m} \lambda_i f_i\Big(y_{j-1}, y_j, \vec{x}, j\Big)\right)
\end{aligned}$$

which means that modeling $\mathcal{Y}_{\vec{x},tu}$ for a piece $\vec{x}$ and a tuning tu allows one to find the same optimal solution $\vec{y}^*$ on a much smaller set of potential solutions than $\mathcal{Y}$. f_{cons} can be omitted because:

$$\forall \vec{y} \in \mathcal{Y}_{\vec{x},tu} : \forall j \in [1,n] : f_{\text{cons}}\Big(y_{j-1}, y_j, \vec{x}, j\Big) = 0$$

Modeling $\mathcal{Y}_{\vec{x},tu}$ by an SFSA creates a structure similar to the layered graphs described by Sayegh [25]: the states of the automaton are structured in layers S_j for

$$S_j = \left\{ s_{j,y} \,\middle|\, y \in \mathit{PLAYABLE\_CONSISTENT}_{\vec{x}_j,tu} \right\}$$

with S_1 being the designated set of initial states and S_n being the set of final states. If a chord $x \in \mathfrak{X}$ occurs multiple times throughout $\vec{x}$, every occurrence is represented by its own layer at the respective position j in $\vec{x}$. Edges connect nodes of neighboring layers:

$$E = \bigcup_{j=2}^{n} \left\{ (s_{j-1,y}, s_{j,y'}) \,\middle|\, s_{j-1,y} \in S_{j-1}, s_{j,y'} \in S_j \right\}$$

The transition probabilities between states are defined as in section 4.1. Every state sequence between an initial state and a final state is a playable fingering $\vec{y}$ consistent to $\vec{x}$ with respect to tu. Creating such an SFSA is simple given a technique for obtaining the sets $\mathit{PLAYABLE\_CONSISTENT}_{\vec{x}_j,tu}$ such as the constraint satisfaction approach described in section 5.1.3. The process of generating a fingering can be canceled immediately in case $\vec{x}$ contains an unplayable chord $\check{x}$ for which $\mathit{PLAYABLE\_CONSISTENT}_{\check{x},tu} = \emptyset$.

Overall, the reformulation via $\mathcal{Y}_{\vec{x},tu}$ has the following advantages:

- It renders the use of CRFs feasible for the guitar fingering problem by reducing the number of states to a manageable amount.
- Forms encountered at any step j during the forward-backward or Viterbi algorithm are known to create the chord $\vec{x}_j$. Contrary to the works of Hori et al. [9], no comparisons are necessary to ensure consistency between forms and chords. This leaves the CRF features $f_i \in \mathcal{F}$ with the only task of characterizing the difficulty of guitar playing.
- Any form in a generated fingering is guaranteed to be biomechanically playable.

Inference

Given a linear-chain CRF with features $f_i \in \mathcal{F}$ characterizing the difficulty of playing and transitioning between forms, their weights $\vec{\lambda}$, a guitar tuning $tu \in \mathbb{P}^6$, find the optimal fingering $\vec{y}^*$ for a piece of one voice $\vec{x} \in \mathfrak{X}^n$:

1. For $j = 1, \ldots, n$, obtain the sets $\mathit{PLAYABLE\_CONSISTENT}_{\vec{x}_j,tu}$ for the chords $\vec{x}_j$ and tu and create an SFSA of a layered structure.
2. Find $\vec{y}^*$ by applying the Viterbi algorithm on the SFSA. In order to exploit the layer structure, the recursion step is changed to:

$$\begin{aligned} \forall s' \in S_j : \quad \delta_j(s') &= \max_{s \in S_{j-1}} \delta_{j-1}(s) \cdot \Psi_j(\vec{x}, s, s') \\ \psi_j(s') &= \operatorname*{argmax}_{s \in S_{j-1}} \delta_{j-1}(s) \cdot \Psi_j(\vec{x}, s, s') \end{aligned}$$

Training

Given a corpus of labeled training data $\mathcal{T} = \left\{ (\vec{x}, \vec{y}, tu) \,\middle|\, \vec{x} \in \mathfrak{X}, \vec{y} \in \mathfrak{Y}, tu \in \mathbb{P}^6 \wedge |\mathcal{Y}_{\vec{x},tu}| \geq 1 \right\}$, features $f_i \in \mathcal{F}$ characterizing the difficulty of playing and transitioning between forms and the regularization constant σ^2, obtain the optimal feature weights $\vec{\lambda}^*$:

1. For each dataset in $\mathcal{T}$, create an SFSA of a layered structure with respect to the dataset's tuning tu.
2. Employ numerical optimization to maximize the log-likelihood on the training data $\mathcal{L}(\mathcal{T})$ to obtain $\vec{\lambda}^*$. Invocations of the forward-backward algorithm run on the respective SFSA for each dataset. A similar modification to the forward-backward algorithm is necessary to exploit the layer structure.

5.3 Application on Guitar Pieces of Multiple Voices

Guitar pieces may be notated as consisting of multiple voices, where each voice exhibits notes of different note values played on different beats. A common example is the separation of melody notes and bass accompaniment into two voices. The possible ways in which the notes of multiple voices can overlap include chords, melodies accompanied by chords and polyphonic textures (the `CHO` and `MIX` patterns and the unsupported pattern described by Radicioni et al. [19]). A (somewhat academic) piece of two voices with accompanied melody and polyphonic textures is shown in figure 8 and is used as an example throughout this section.

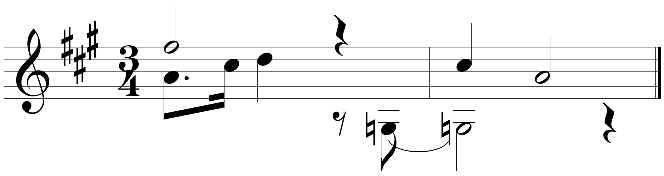


Figure 8.: Example piece of two voices (notated in one staff)

The challenge in generating fingering for pieces of multiple voices is to resolve the dependencies introduced by the overlapping notes: The assignment of a certain fingered position to a note held for the duration of one bar could lead to a dead end several notes later because no biomechanically playable forms remain, given the assignment.

CRF Definition

Before modeling the CRF, the representation of guitar pieces needs to be revised as overlapping notes cannot be modeled by a sequence of self-contained chords. Instead, notes need to be associated with their start time and end time within a piece, for which the unit of time can be chosen arbitrarily. In this thesis, points in time are represented by the elapsed microseconds since the start of a piece.

Another reformulation is necessary before one can specify the observations and labels of the CRF: Given a piece of multiple voices, flatten the piece to one voice. This allows one to represent the overlapping notes of multiple voices as a sequence of chords with tied notes, where the notes within each chord are of the same note value[1)]. The reformulated version of the example piece is shown in figure 9.

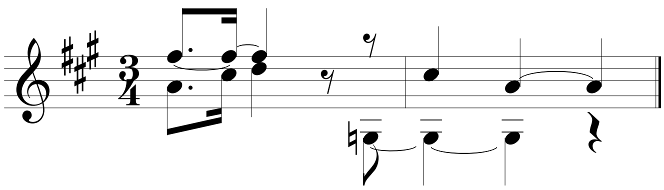


Figure 9.: Example piece flattened from two voices to one voice

The resulting chords are named *constant segments* in this thesis because they are the segments of a guitar piece in which the set of currently played notes remains constant. From one segment to the next, notes supervene, cease or persist. The set of constant segments *CS* is defined as

$$CS = \left\{ \left(\vec{p}, \vec{tied}, t_{\text{start}}, t_{\text{end}}\right) \,\middle|\, \vec{p} \in \mathbb{P}^c,\, c \in [1{,}6],\, \vec{tied} \in \{true, false\}^c,\, t_{\text{start}} \in \mathbb{N},\, t_{\text{end}} \in \mathbb{N} \right\}$$

1) This does not change the acoustic outcome of a piece: Tied notes are not played another time but are merely a notation for extending the duration of the previous note.

with one constant segment $cs \in CS$ being a 4-tuple of a list of pitches $\vec{p}$, a list of truth values $\vec{tied}$ denoting whether the i-th note of a segment is tied from the previous segment and the points in time $t_{\text{start}}, t_{\text{end}}$. The constant segments of the example piece are listed in table 2. Forms can be assigned to the pitches of constant segments the same way they can be assigned to the chords of pieces of one voice.

	$\vec{p}$	$\vec{tied}$	t_{start}	t_{end}
$\vec{x}_1 =$	$\big($(A3, F♯4),	$(false, false)$,	0,	375 000$\big)$
$\vec{x}_2 =$	$\big($(C♯4, F♯4),	$(false, true)$,	375 000,	500 000$\big)$
$\vec{x}_3 =$	$\big($(D4, F♯4),	$(false, true)$,	500 000,	1 000 000$\big)$
$\vec{x}_4 =$	$\big($(G2),	$(false)$,	1 250 000,	1 500 000$\big)$
$\vec{x}_5 =$	$\big($(G2, C♯4),	$(true, false)$,	1 500 000,	2 000 000$\big)$
$\vec{x}_6 =$	$\big($(G2, A3),	$(true, false)$,	2 000 000,	2 500 000$\big)$
$\vec{x}_7 =$	$\big($(A3),	$(true)$,	2 500 000,	3 000 000$\big)$

Table 2.: Constant segments of the example piece (assuming a tempo of 120 bpm)

The labels and observations of the CRF are therefore defined as

$$\mathfrak{Y} = PLAYABLE$$
$$\mathfrak{X} = CS$$

with the goal of finding the most likely sequence of forms to a piece given as a sequence of constant segments.

The next task is to model $\mathcal{Y}_{\vec{x},tu}$ by an SFSA which respects the dependencies between constant segments. If a note is split up across several segments, it must be assigned the same fingered position in every one of these segments. This can be expressed by a predicate over two subsequent segments $\vec{x}_{j-1}, \vec{x}_j \in CS$ and their corresponding forms $y \in PLAYABLE\_CONSISTENT_{\vec{x}_{j-1},tu}$, $y' \in PLAYABLE\_CONSISTENT_{\vec{x}_j,tu}$ defined as

$$\text{match}(\vec{x}_{j-1}, \vec{x}_j, y, y') \Longleftrightarrow \forall i \in \big[1, |y'|\big] : \text{tied}(\vec{x}_j)_i = true \Longrightarrow \exists k \in \big[1, |y|\big] : \text{p}(\vec{x}_{j-1})_k = \text{p}(\vec{x}_j)_i \wedge y_i = y'_k$$

In words: For every index i indicating a tied note within $\vec{x}_j$, an index k has to exist which indicates the same pitch within $\text{p}(\vec{x}_{j-1})$. The fingered positions of each tied note, located at i and k in the forms for $\vec{x}_j$ and $\vec{x}_{j-1}$ respectively, must match.

The process of creating an SFSA starts out with

$$S_1 = \left\{ s_{1,y} \;\middle|\; y \in PLAYABLE\_CONSISTENT_{\vec{x}_1,tu} \right\}$$

and continues for $j = 2, \ldots, n$ with

$$S_j = \left\{ s_{j,y'} \;\middle|\; y' \in PLAYABLE\_CONSISTENT_{\vec{x}_j,tu} \wedge \exists\, s_{j-1,y} \in S_{j-1} : \text{match}(\vec{x}_{j-1}, \vec{x}, y, y') \right\}$$
$$E_{j-1,j} = \left\{ (s_{j-1,y}, s_{j,y'}) \;\middle|\; s_{j-1,y} \in S_{j-1}, s_{j,y'} \in S_j \;: \text{match}(\vec{x}_{j-1}, \vec{x}_j, y, y') \right\}$$

As the SFSA is created, the match predicate is enforced for new states so that any state added to layer S_j is guaranteed to have at least one matching predecessor state from S_{j-1}. However, this procedure cannot guarantee that at least one matching successor from S_j exists for every state in S_{j-1}. States without a

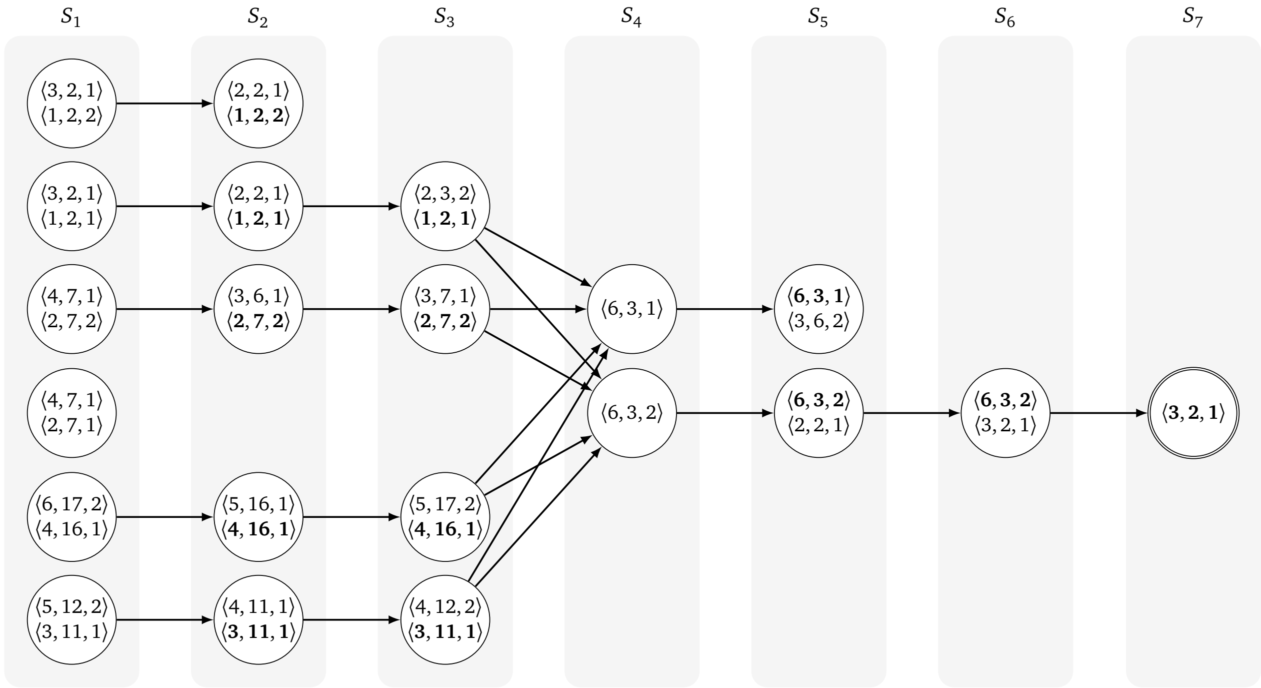


Figure 10.: Layered SFSA of the example piece including dead ends – emphasized: fingered positions of tied notes. The SFSA was created for standard tuning and with the omission of ring and little finger to keep the number of states manageable. Notice how the reason for failure in the topmost row dates back to a decision taken two segments earlier.

successor correspond to the aforementioned dead ends where no biomechanically playable forms remain. Figure 10 shows the SFSA of the example piece including several dead ends.

Dead ends can be removed by iterating backwards over the segments $S_{n-1}, \ldots, S_1$ and removing any state $s_{j,y}$ not connected to any successor $s_{j+1,y'} \in S_{j+1}$. Once dead ends are removed, all paths through the SFSA represent consistent and biomechanically playable fingerings for $\vec{x}$, meaning the SFSA represents $\mathcal{Y}_{\vec{x},tu}$.

The SFSA of the example piece after removing the dead ends is shown in figure 11.

Inference and Training

With the knowledge on how to obtain the SFSA for pieces of multiple voices one can turn toward inference and training. Changing the definition of the observations $\mathfrak{X}$ requires no major modification to the procedures given in section 5.2 for pieces of one voice. The features $f_i \in \mathcal{F}$ again characterize the difficulty of playing and transitioning between forms. The timing information included in the constant segments can be incorporated in the feature calculations.

An important change regarding the training procedure is the format of the training data which needs to be supplied in the form of labeled constant segments with timing information. This complicates the acquisition of training data because in a addition to the reference fingering, a machine-readable representation of note values is required (for example in the form of a MIDI file). Automatic extraction of training data from ASCII tablature (as done by Tuohy et al. [32]) is not feasible because there is no standard way of notating timing information in ASCII tablature.

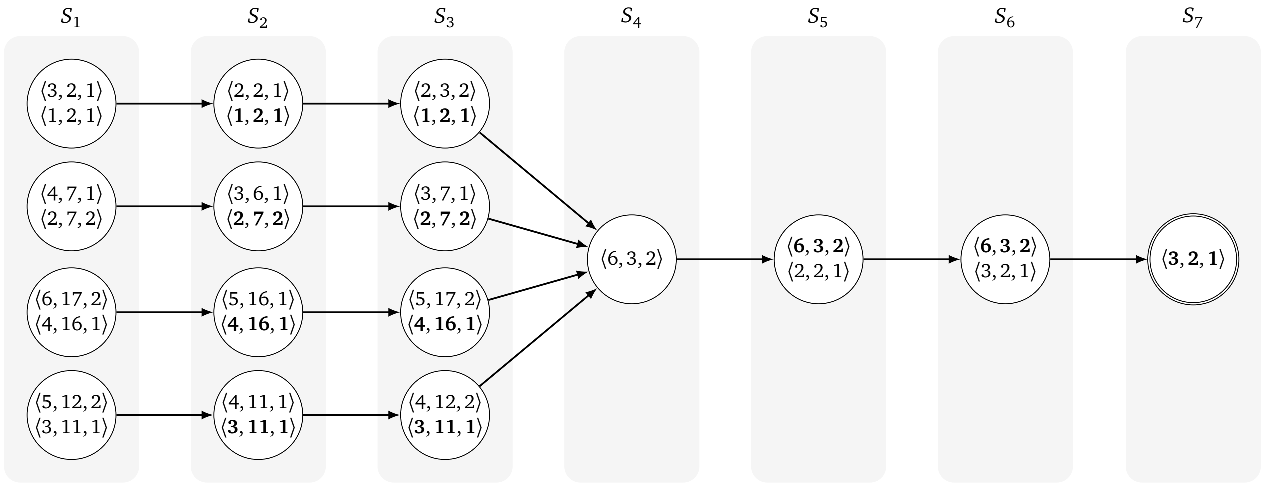


Figure 11.: Final layered SFSA of the example piece without dead ends – emphasized: fingered positions of tied notes.

5.4 Feature Definitions

Each component of the cost functions found in published literature (e. g. state transition probabilites, fitness functions or others) can be categorized according to table 3. A component either targets the whole hand or individual fingers, judges transitions between forms or single forms and, in a broader sense, deals with the biomechanical or cognitive difficulty of playing the guitar.

	Cognitive	Biomechanical	
		Static	Transitional
Hand	position along the fretboard	comfort of forms	hand movement between forms
Fingers	complexity and reuse of forms	comfort of forms and fret position (w. r. t force)	finger movement between forms

Table 3.: Categorization of cost function components by body part and difficulty aspects of guitar play

The following sections describe the CRF features implemented in this thesis, grouped by their category. Beforehand, limitations and aspects applying to all CRF features are stated.

5.4.1 Limitations of CRF Features

Due to the first-order Markov assumption, CRF features are based only on two forms y_{j-1}, y_j and the constant segments of a piece without any context of previous forms. Higher-order difficulty aspects such as the reuse of forms within a piece cannot be implemented and have to be excluded altogether.

The amount of information available on finger positions available to CRF features is sparse because forms only model the positions of fingers which actively depress strings. This issue especially affects features concerning the position of the left hand: As stated in section 2.2, the position of the hand along the fretboard hinges on the position of the index finger, whose position is estimated if it does not depress any strings in a form. This estimation fails for forms consisting only of open strings where there are no fingers available as a reference. In these cases, the hand could be located anywhere on the fretboard.

Consider the following two examples:

1. A sequence of three forms where the first form is played in the first fret, the second form consists only of open strings and the third form is played in the first fret again.
2. A sequence of three forms where the first form is played in the first fret, the second form consists only of open strings and the third form is played in the fifth fret.

In the first example, guitarists would keep their hand positioned in the first fret while playing the second form. In the second example, a transition from the first to the fifth fret is necessary. Guitarists would take advantage of the fact that the left hand is not used in the second form by already moving their hand during the second form, thereby lengthening the time period available for the transition.

The two examples are indistinguishable to a CRF feature which has only access to two forms at once. The most reasonable way of dealing with this limitation given the forms y_{j-1} and y_j is to assume that the hand stays in place when y_j is a form consisting only of open strings. In case y_{j-1} was a form consisting only of open strings, assume that the hand is already positioned in the fret of y_j. In the special case that both y_{j-1} and y_j are only played on open strings, assume again that no hand movement along the fretboard is necessary.

Common Aspects among CRF Features

Within the scope of the CRF feature definitions, let

$$fr_{\text{index}} : \mathit{PLAYABLE} \rightarrow [-2, 19] \cup \lozenge$$

denote the function returning the fret of the index finger in a form $\vec{pos} \in \mathit{PLAYABLE}$ or, if the index finger is not active in the form, its estimated fret. $\lozenge$ is returned when $\vec{pos}$ consists only of open strings.

Also needed is a function which returns the metric distance between the nut and fret fr, given a scale length s (see section 2.1). The function is defined as

$$\text{nut\_dist} : [-2, 19] \rightarrow \mathbb{R}[\text{mm}]$$
$$\text{nut\_dist}(fr) = s \cdot \left(1 - 2^{-\frac{fr}{12}}\right)$$

for the scale length $s = 650\,\text{mm}$ of a classical guitar.

Every feature should have the same co-domain so that the regularization operates correctly during training (see section 4.3). The co-domain $[0, 1]$ is the most reasonable choice because of its universal applicability to other approaches including other probabilistic models such as HMMs. Most features are based on the hyperbolic function $\frac{1}{1+x}$ for this reason.

5.4.2 Cognitive Features Targeting the Hand

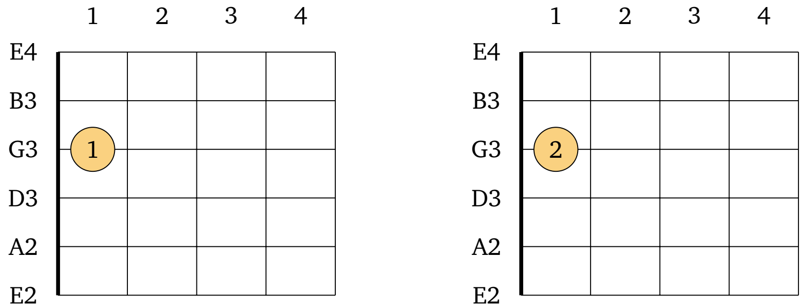
Fretwise hand position

This feature judges the hand position along the neck in relation to the fret in which the index finger resides and is adopted from Hori et al. [9]. It penalizes playing in higher frets with regard to the findings of Heijink et al. [7].

The original formulation found in [9] is:

$$f\left(y_{j-1}, y_j, \vec{x}, j\right) = \frac{1}{1 + fr_{\text{index}}(y_j)}$$

Hori et al. [9] do not mention the special case of forms consisting only of open strings discussed above. Apart from this imprecision, the formulation overshoots its target at the lower frets: Because the highest probability is achieved by $fr_{\text{index}}(y_j) = 0$, forms located "in the zeroth fret" are preferred over forms played in the first fret (see figure 12). Furthermore, playing at fret −1 leads to a division by zero which is also not addressed by Hori et al.[2)] These two flaws might not have arisen depending on how the forms were specified for their HMM.



(a) index finger located in fret 1 **(b)** index finger located in fret 0

Figure 12.: Fretboard charts of forms differing in the location of the index finger

This thesis uses a formulation which mitigates the division by zero and the preference for playing in the negative fret range:

$$f_{\text{hand\_fret}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \begin{cases} 1 & \text{if } fr_{\text{index}}(y_j) = \Diamond \\ \dfrac{1}{|fr_{\text{index}}(y_j)| + 2} & \text{if } fr_{\text{index}}(y_j) \leq 0 \\ \dfrac{1}{fr_{\text{index}}(y_j)} & \text{otherwise} \end{cases}$$

As a result, forms consisting only of open strings receive the value 1. The value for forms in the standard range $[1,19]$ starts with 1 and decreases in hyperbolic fashion with increasing frets. Forms located in the frets $\{0, -1, -2\}$ receive the same values as forms in the frets $\{2, 3, 4\}$.

Hand positioned in the first fret or higher

Even with the removed preference for negative frets (see above), unusual finger assignments for positions in low frets were observed in test runs. At times, the little finger was assigned to positions in the first fret without necessity. It was therefore decided to introduce a boolean feature which encodes whether the hand is positioned in the first fret or higher.

$$f_{\text{hand\_first}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \begin{cases} 1 & \text{if } fr_{\text{index}}(y_j) = \Diamond \vee fr_{\text{index}}(y_j) \geq 1 \\ 0 & \text{otherwise} \end{cases}$$

5.4.3 Cognitive Features Targeting the Fingers

Complex forms involving many fingers lead to a higher cognitive effort than forms of few fingers: On the one hand, guitarists need to memorize a larger number of positions for each finger. On the other hand, transitions between forms grow more difficult because guitarists need to move the right fingers among several.[3)]

2) The authors were not available to discuss this issue.

3) This is opposed to the difficulty of executing the transition, which is of biomechanical nature.

Number of fingers involved in a form

This feature penalizes forms using many fingers and is adopted from Hori et al. [9] without modifications. Given a form $\vec{pos} \in PLAYABLE$, define $FINGERS_{\vec{pos}}$ as the set of fingers used in $\vec{pos}$ excluding finger 0 which is only assigned to open strings.

The feature is defined as:

$$f_{\text{number\_of\_fingers}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \frac{1}{1 + |FINGERS_{y_j}|}$$

Number of fingers added / moved / removed / changed between two forms

Given two forms $\vec{pos}_1, \vec{pos}_2 \in PLAYABLE$, define the sets:

- $ADDED_{\vec{pos}_1,\vec{pos}_2}$: The set of fingers appearing in $\vec{pos}_2$ but not in $\vec{pos}_1$, excluding finger 0.
- $REMOVED_{\vec{pos}_1,\vec{pos}_2}$: The set of fingers appearing in $\vec{pos}_1$ but not in $\vec{pos}_2$, excluding finger 0.
- $MOVED_{\vec{pos}_1,\vec{pos}_2}$: The set of fingers appearing in $\vec{pos}_2$ and $\vec{pos}_1$ whose position(s) changed. Keep in mind that the index finger can play multiple positions by means of a barre chord.

The features are defined as:

$$f_{\text{added}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \frac{1}{1 + |ADDED_{y_{j-1},y_j}|}$$

$$f_{\text{removed}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \frac{1}{1 + |REMOVED_{y_{j-1},y_j}|}$$

$$f_{\text{moved}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \frac{1}{1 + |MOVED_{y_{j-1},y_j}|}$$

$$f_{\text{changed}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \frac{1}{1 + |ADDED_{y_{j-1},y_j}| + |REMOVED_{y_{j-1},y_j}| + |MOVED_{y_{j-1},y_j}|}$$

5.4.4 Static Features Targeting the Hand

Fretwise width of a form

The goal of this feature is to penalize forms with a high fret span. It was adapted from Hori et al. [9] without modifications.

The fretwise width of a form $\vec{pos} \in PLAYABLE$ is returned by the function

$$\text{width}(\vec{pos}) = \begin{cases} 0 & \text{if } fr_{\text{index}}(\vec{pos}) = \Diamond \\ (\max_{fr} \vec{pos}) - \big((\min_{fr} \vec{pos}) - 1\big) & \text{otherwise} \end{cases}$$

The feature is defined as:

$$f_{\text{width\_fretwise}}\big(y_{j-1}, y_j, \vec{x}, j\big) = \frac{1}{1 + \text{width}(y_j)}$$

Metric width of a form

An issue of the previous feature is the fact that it disregards the actual fret spacing found on a guitar: A form spanning five frets would result in the same value regardless of whether it is played in the first or in the twelfth fret, even though the lower fret spacing around the twelfth fret renders the latter form more comfortable to play. This feature is supposed to be an improvement over the previous feature by judging the width of a form with respect to the fret spacing.

The width of a form is redefined relative to the width of the first fret as:

$$\text{metric\_width}(\vec{pos}) = \begin{cases} 0 & \text{if } fr_{\text{index}}(\vec{pos}) = \Diamond \\ \dfrac{\text{nut\_dist}(\max_{fr} y_j) - \text{nut\_dist}((\min_{fr} y_j) - 1)}{\text{nut\_dist}(1)} & \text{otherwise} \end{cases}$$

The feature value is computed by

$$f_{\text{width\_metric}}\big(y_{j-1}, y_j, \vec{x}, j\big) = \frac{1}{1 + \text{metric\_width}(y_j)}$$

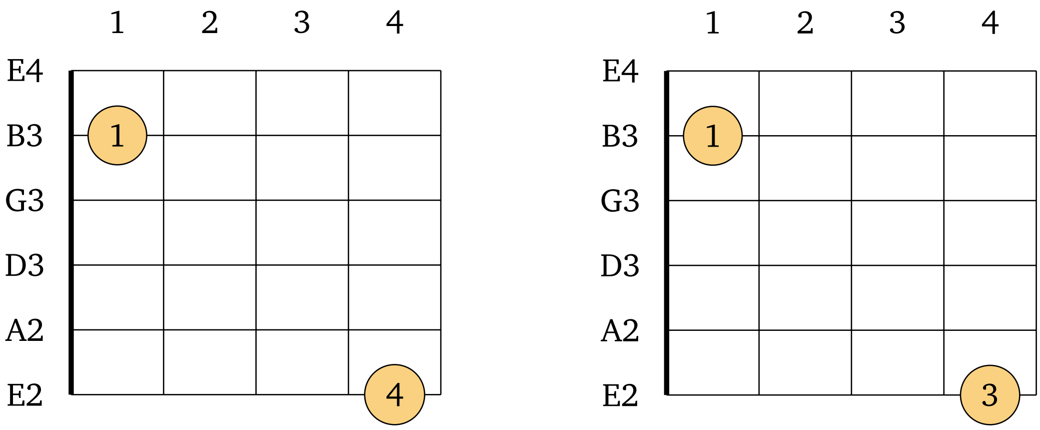
Hand stretching feature

The previously described feature comes closer to modeling reality but is still flawed with respect to the fingers involved in a form. Figure 13 shows two forms which receive the same feature value because they cover the same frets. Most guitarists would consider the first form to be easier to play because no stretching is necessary to reach the positions on the fretboard.

A feature judging the comfort of the left hand should therefore respect the fingers assigned to the positions which affect the width of a form. The hand stretching feature employs the proportion of a performer's hand breadth to the metric width of a form for this purpose.

Going back to form (a) in figure 13, the span of the fretboard actually covered by the left hand is the width of the frets 2, 3 and 4 plus the width of a performer's index finger. The width of the first fret needs not to be included because guitarists place their fingers close to the frets [7][4].

[4] This is a simplified statement. Heijink et al. [7] report that performers actually divert from this principle the lower the hand is positioned on the fretboard. It is nonetheless preferable to position the fingers close to the frets to avoid an unwanted buzzing sound when playing a string.



(a) no stretching required

(b) stretching required

Figure 13.: Fretboard charts of forms differing with respect to hand stretching

The closest information available on the hand dimensions of guitarists is a study conducted by Wagner [34] covering the variation in hand dimensions across male and female pianists. Wagner's results should be roughly comparable. The median of the breadth of the left hand measured at the finger pads is reported to be 84 mm for male subjects and 76 mm for female subjects. The average breadth of 80 mm was chosen for computations in this thesis. Measurements on finger breadth are not included in the study. The breadth of a finger was chosen to be 13 mm, originating from the fret distance found on a classical guitar.

Using these measurements, the proportion of a performer's hand breadth to the metric width of the form (a) in figure 13 becomes:

$$\frac{80\,\text{mm}}{13\,\text{mm} + \text{nut\_dist}(4) - \text{nut\_dist}(1)} = \frac{80\,\text{mm}}{110.6\,\text{mm}} = 0.723$$

The metric width remains the same for the second, more difficult form (b). However, because the index finger is located in the first fret and the ring finger is located in the fourth fret, three of the four fingers of the left hand need to cover the same span of 110.6 mm on the fretboard. Consequently, the proportion should be defined as:

$$\frac{\frac{3}{4} \cdot 80\,\text{mm}}{13\,\text{mm} + \text{nut\_dist}(4) - \text{nut\_dist}(1)} = \frac{60\,\text{mm}}{110.6\,\text{mm}} = 0.542$$

The generalized formulation of the hand stretching feature, given a hand breadth h and a finger breadth b is:

$$f_{\text{stretch}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \begin{cases} 1 & \text{if } fr_{\text{index}}(y_j) = \Diamond \\ \min\left(1; \dfrac{\frac{1}{4} \cdot \left(1 + \max_{fi} y_j - \min_{fi} y_j\right) \cdot h}{b + \text{nut\_dist}(\max_{fr} y_j) - \text{nut\_dist}(\min_{fr} y_j)}\right) & \text{otherwise} \end{cases}$$

The proportion exceeds 1 for forms using many fingers in few frets. The feature value was capped at 1 to conform to the uniform feature co-domain for these cases.

For the sake of comparability, the hand stretching feature can also be realized in a simplified version based on distances measured in frets. In this case, the feature judges the proportion between the range of fingers involved in a form and its fretwise width. Using the width function defined for $f_{\text{width\_fretwise}}$, the simplified version is given by:

$$f_{\text{stretch\_fretwise}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \begin{cases} 1 & \text{if } fr_{\text{index}}(y_j) = \lozenge \\ \min\left(1; \dfrac{\max_{fi} y_j - \min_{fi} y_j + 1}{\text{width}(y_j)}\right) & \text{otherwise} \end{cases}$$

5.4.5 Static Features Targeting the Fingers

Maximum number of fingers in a fret

From personal experience, forms with many fingers positioned close to each other within a fret (such as the A major form depicted in figure 14) can be difficult to play.

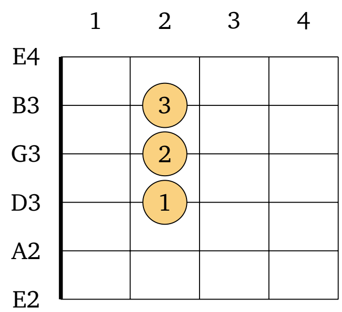


Figure 14.: Fretboard chart of a common A major form

Forms of this kind can be penalized by a feature which counts the maximum number of fingers occurring in one fret of a form. This is achieved by the auxiliary function

$$\text{max\_fingers\_per\_fret}(\vec{pos}) = \max_{fr=1,\ldots,19} |\{ fi \mid \langle s, fr, fi \rangle \in \vec{pos} \}|$$

Using this function, the feature definition is given by:

$$f_{\text{max\_fingers}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \frac{1}{1 + \text{max\_fingers\_per\_fret}(y_j)}$$

A shortcoming of this formulation is that apart from counting fingers in the same fret, the number of strings between fingers is not considered. Forms with three fingers in one fret with each finger positioned one string apart are unjustly penalized.

5.4.6 Transitional Features Targeting the Hand

Hand movement along the fretboard over time (Laplace)

Another feature adopted from the works of Hori et al. [9] is a feature judging the movement of the left hand along the fretboard. Its general principle was already described in the literature review in section 3.1.

The cost of hand movements along the fretboard is modeled by a Laplace distribution whose variance is proportional to the time interval available for the transition. As a result, the Laplace distribution approaches the uniform distribution the larger the time interval becomes, meaning far movements are tolerated if there is enough time available [9].

The formula stated by Hori et al. is:

$$f\left(y_{j-1}, y_j, \vec{x}, j\right) = \frac{1}{2\Delta t(\vec{x}, j)} \cdot \exp\left(\frac{-|fr_{\text{index}}(y_{j-1}) - fr_{\text{index}}(y_j)|}{\Delta t(\vec{x}, j)}\right)$$

The time interval Δt is defined as the interval between the start of two chords (two constant segments in the context of this thesis):

$$\Delta t(\vec{x}, j) = \text{t}_{\text{start}}(\vec{x}_j) - \text{t}_{\text{start}}(\vec{x}_{j-1})$$

Hori et al. [9] make no mention of how forms consisting only of open strings were treated. The unit of Δt is not reported. Note that the definition of Δt counts the time in which notes are sustained as being available for hand movement too, effectively reducing the duration of each note to zero.

Experimentation with the formula revealed that using hundredths of a second as the unit of Δt produces the best results. The formulation chosen for the hand movement feature based on the Laplace distribution is:

$$f_{\text{hand\_laplace}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \begin{cases} 1 & \text{if } fr_{\text{index}}(y_{j-1}) = \Diamond \vee \\ & \quad fr_{\text{index}}(y_j) \quad = \Diamond \\ \frac{1}{2\Delta t(\vec{x}, j)} \cdot \exp\left(\frac{-|fr_{\text{index}}(y_{j-1}) - fr_{\text{index}}(y_j)|}{\Delta t(\vec{x}, j)}\right) & \text{otherwise} \end{cases}$$

for

$$\Delta t(\vec{x}, j) = \text{t}_{\text{start}}(\vec{x}_j) - \text{t}_{\text{start}}(\vec{x}_{j-1})$$

Hand movement along the fretboard over time (Fitts)

Another approach for judging the difficulty of hand movements over a period of time can be derived from Fitts's law.

Fitts's law is commonly used in the area of human-computer interaction to evaluate pointing actions [5, 6]. It specifies the time necessary to point on a target in relation to the distance to the target and the target size [5]. The law was expressed by several different formulas over the years [5, 8] of which the formula closest to the original publication by Fitts [6] was chosen in this thesis. The expected time t is defined as

$$t = a + b \cdot \log_2\left(\frac{2A}{W}\right)$$

given the initial distance to the target A, the target width W, the initial reaction time a and the index of performance $\frac{1}{b}$. The index of performance is the ratio between the difficulty of a task and the available time t [6]. Consequently, it increases for larger target distances and lower time periods.

An application of Fitts's law to the guitar fingering problem was mentioned by Allen et al. [1] but was not covered in detail. For this feature, the task of moving the left hand along the fretboard to a certain fret is interpreted as a one-dimensional pointing task with the intent of expressing the difficulty of hand movements by the index of performance $\frac{1}{b}$.

t corresponds to $\Delta t(\vec{x}, j)$, the time available for moving the hand as defined in the previous feature. The target distance A is given by:

$$A = \left|\text{nut\_dist}\left(fr_{\text{index}}(y_j)\right) - \text{nut\_dist}\left(fr_{\text{index}}(y_{j-1})\right)\right|$$

The width of the target W is defined as the size of a fingertip which is again defined as 13 mm, the smallest fret distance found on a classical guitar.

First, rearrange Fitts's law to

$$t = a + b \cdot \log_2\left(\frac{2A}{W}\right)$$
$$\iff \frac{1}{b} = \frac{\log_2\left(\frac{2A}{W}\right)}{t-a}$$

Once a guitarist has sufficiently practiced a piece, movements of the left hand become planned actions instead of reactions. One can therefore set the initial reaction time to $a = 0$.[5)]

The index of performance is then inserted into the hyperbolic function $\frac{1}{1+x}$ to ensure a feature domain of $[0,1]$:

$$\frac{1}{1+\frac{1}{b}} = \frac{1}{1+\frac{\log_2\left(\frac{2A}{W}\right)}{t}} = \frac{t}{t+\log_2\left(\frac{2A}{W}\right)}$$

The most suitable unit of t found for the hyperbolic formulation was again hundredths of a second. In theory, the feature value could exceed 1 for $\frac{1}{b} < 0$, which happens if the target distance is less than the target size, meaning the target is already in reach. Since the fingertip size (the target size) was defined as the smallest possible fret distance (the minimum possible target distance), this special case cannot occur in this feature.

The complete definition of the feature is:

$$f_{\text{hand\_fitts}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \begin{cases} 1 & \text{if } fr_{\text{index}}(y_{j-1}) = \lozenge \lor fr_{\text{index}}(y_j) = \lozenge \\ \dfrac{t}{t+\log_2\left(\frac{2A}{W}\right)} & \text{otherwise} \end{cases}$$

for

$$\begin{aligned} \Delta t(\vec{x}, j) &= \mathrm{t_{start}}(\vec{x}_j) - \mathrm{t_{start}}(\vec{x}_{j-1}) \\ A &= \left|\text{nut\_dist}\left(fr_{\text{index}}(y_j)\right) - \text{nut\_dist}\left(fr_{\text{index}}(y_{j-1})\right)\right| \\ W &= 13 \end{aligned}$$

An simplified formulation $f_{\text{hand\_fitts\_fretwise}}$ of the feature relying on fretwise distances can be defined by replacing A and W by

$$\begin{aligned} A &= \left|fr_{\text{index}}(y_j) - fr_{\text{index}}(y_{j-1})\right| \\ W &= 1 \end{aligned}$$

Hand repositioning necessary

This feature is a simple boolean feature which returns 0 whenever the hand needs to move between two forms. It is defined as:

$$f_{\text{hand\_reposition}}\left(y_{j-1}, y_j, \vec{x}, j\right) = \begin{cases} 1 & \text{if } fr_{\text{index}}(y_{j-1}) \neq \lozenge \land fr_{\text{index}}(y_{j-1}) = fr_{\text{index}}(y_j) \\ 0 & \text{otherwise} \end{cases}$$

[5)] Values > 0 could be used to generate suitable fingerings for sight-reading, where a performer plays a piece ad hoc from sheet music without previously practicing the piece.

5.4.7 Features Covering Multiple Categories

HMM state transition probability by Hori et al. [9]

A feature computing the state transition probability employed in the works of Hori et al. [9] was implemented for the sake of comparability. It is defined as a multiplication of several features:

$$f_{\text{hmm}}\left(y_{j-1}, y_j, \vec{x}, j\right) = f_{\text{hand\_laplace}}\left(y_{j-1}, y_j, \vec{x}, j\right) \cdot f_{\text{hand\_fret}}\left(y_{j-1}, y_j, \vec{x}, j\right) \cdot f_{\text{width\_fretwise}}\left(y_{j-1}, y_j, \vec{x}, j\right) \cdot f_{\text{number\_of\_fingers}}\left(y_{j-1}, y_j, \vec{x}, j\right)$$

6 Implementation and Feature Selection

6.1 Implementation

The approach to solving the guitar fingering problem presented throughout the previous chapters was implemented in the Java programming language [16].

The linear-chain CRF was implemented using a publicly available CRF library [26] which was customized to support the presented layered SFSAs in the forward-backward and Viterbi algorithms.

The constraint satisfaction problem responsible for finding the sets $PLAYABLE\_CONSISTENT_{\vec{p},tu}$ was modeled and solved with the choco-solver library [17], a free and open source library specializing on constraint programming. Because choco-solver found solutions to the CSPs in satisfactory time running on the default settings, the settings were left unchanged.

Guitar pieces were supplied to the system in the form of MIDI files. In a MIDI file, each note is encoded by two events denoting the start and end time within the piece [2]. This representation allows one to directly segment a guitar piece into constant segments without much preprocessing. Each note in a MIDI file belongs to a *channel*, which themselves belong to *tracks*. Unisons (two or more notes of the same pitch played in a chord) need to be encoded by playing two notes of the same pitch on different channels or different tracks. The implementation in this thesis does not differentiate between the channel or track a note originated from, meaning unisons are not supported. Naturally, a different representation than MIDI could be used as an input format as long as it provides the notes of a piece with their start and end time.

The reference fingering of each training piece was stored as a CSV file. The contents of such a file are a reference form for each constant segment of a piece, addressed by the start time of the constant segment t_{start}. Guitar tunings of training pieces were represented by serialized Java objects.

Given a set of CRF features, a set of training data (one MIDI file, CSV file and tuning file per piece) and the dimensions of a guitarist and their instrument, the system returns the corresponding feature weights $\vec{\lambda}$ obtained by training the CRF on the data. The best fingering to a guitar piece can be inferred by providing the piece as a MIDI file along with a guitar tuning, a set of CRF features, their weights $\vec{\lambda}$ and the dimensions of performer and instrument. The implemented output modalities for generated fingerings are ASCII tablature and optionally a CSV file of the above format for further analysis.

Unit tests were employed as a countermeasure against programming mistakes in central components of the implementation such as the creation of constant segments from MIDI files and the CRF feature computations.

6.2 Feature Selection

Features of low explanatory power require processing time during inference and training but hardly contribute to $p(\vec{y}|\vec{x})$. It is therefore beneficial to determine the subset of features with high explanatory power among those listed in section 5.4.

The selection was realized by greedy forward selection. This technique is first explained for the general case before moving on to its application on the CRF.

6.2.1 General Selection Procedure

The general procedure for a given collection of features $\mathcal{F}$ and labeled data T is [35, p. 292]:

1. Start with an empty feature set F.
2. Estimate the accuracy of each feature set $F \cup \{f\}$, $f \in \mathcal{F}$ on the training data T.
3. Redefine F as the feature set which achieved the highest estimated accuracy in step 2.
4. Return F if the termination criterion holds, otherwise continue with step 2.

Possible termination criteria include termination once F reaches a certain number of elements or termination if the accuracy did not increase from one iteration to the next.

The accuracy estimation in step 2 can be realized by k-fold cross validation. Compared to holdout evaluation, where the accuracy is determined by training and testing a model on designated datasets, cross validation is more efficient at reusing labeled data [35, p. 149]. This is advantageous since labeled datasets may be costly to obtain [35, p. 146]. The process is:

1. Randomly divide the training data into k subsets of approximately the same size.
2. Run k iterations where in each iteration, subset number k is put aside as the test set. A model is trained on the remaining $k-1$ subsets using the set of features $F \cup \{f\}$ and is then evaluated on the test set.
3. The total accuracy achieved with $F \cup \{f\}$ is the average of the k accuracies from step 2.

6.2.2 Composition of the Dataset used for Feature Selection and Training

The labeled dataset consisted of 10 manually selected guitar pieces from classtab.org for which MIDI data and ASCII tablature with left-hand fingering was available. The pieces were chosen to represent a mixture of difficulties, tempos, hand positions along the fretboard and styles (more melodic or more chord-like). The tablatures were created by various authors. One piece consisted of significantly more notes than the other pieces and was therefore shortened to half its length to reduce its impact on the feature selection.

Smaller mistakes and inconsistencies in the MIDI files and the fingerings were resolved manually:

- Notes in MIDI files were quantized (note start and end times were moved onto the nearest beat) to eliminate eventual interpretations of the guitarist who recorded the MIDI file and to reduce the number of constant segments.
- Repetitions without variation were removed since they provide no new knowledge.
- Trill notes missing in MIDI files were added with a note value of one quarter of the following note.
- Obvious mistakes in MIDI files and the fingerings were fixed. Common mistakes included wrong notes (wrong with respect to personal assessment and sheet music or videos of professional performances found online), wrong frets or accidentally interchanged finger assignments in the tablature.
- Unsupported sections such as chords including unisonos or barre chords played by fingers other than the index finger were removed or modified to ensure the reference fingering could be generated by the CRF. Certain held notes in MIDI files had to be shortened to ensure the playability of a piece.
- Certain fingered positions of evidently poor quality were modified based on personal opinion.

The chosen pieces and the detailed modifications made to the pieces are listed in appendix A. In total, the labeled dataset consisted of 2897 notes.

After applying the above corrections, a preliminary fingering was generated for each piece in the form of a CSV file. The preliminary fingerings were then manually edited to match the tablature.

6.2.3 Adaptation of the Selection Procedure

Every feature of those defined in section 5.4 was included in the feature selection except for f_{hmm} because its values already stem from a selection of several features.

Longer guitar pieces were split up to homogenize the number of notes in each cross validation fold. Given the lack of higher-order features, the negative consequences of splitting guitar pieces were considered to be negligible. As a result, 15 parts of pieces of roughly 100 to 300 notes each were obtained.

The feature selection approach consisted of five executions of greedy forward selection where the accuracy was estimated by means of cross validation.

In each execution, the pseudorandom number generator responsible for dividing the labeled data into cross validation folds was initialized by a different seed. The division of the data was kept for one entire run of greedy forward selection. Greedy forward selection was run until adding another feature would have lead to a reduction in accuracy compared to the last iteration.

The final feature set was derived from the results of the five executions by picking all those features which were selected in at least two executions.

Cross validation was carried out for $k = 5$, meaning three parts of guitar pieces per fold. In each iteration of cross validation, a CRF was trained on a given feature set using $\sigma^2 = 10$. To estimate the accuracy of the CRF, a fingering was generated for each piece in the test set which was then compared note by note to the reference fingering. The accuracy was defined as the percentage of matching fingered positions (3-tuples of string, fret and finger) between the two. The accuracies were averaged on a per-note basis across the 5 cross validation iterations. Averaging the accuracies per subset would have assigned a higher weight to notes belonging to a fold consisting of shorter pieces.

A note on the reliability and appropriateness of determining the accuracy of a generated fingering: Remember that more than one fingering can be considered optimal for one piece since the optimality condition is of subjective nature for the guitar fingering problem. Comparing a generated fingering to one reference fingering gives the false impression of only one "correct" solution. However, even if multiple optimal fingerings were known for a piece, note by note comparisons could not reward generated fingerings which are internally consistent but deviate from every reference fingering. The theoretically best solution to this problem would be to let a representative sample of guitarists evaluate generated fingerings. This is not a feasible approach, especially not when conducting a feature selection where several thousand fingerings need to be evaluated. Therefore, despite it being only a rough guidance, the percentage of matching fingered positions determined by a note by note comparison is the best choice for a feature selection.

6.2.4 Selection Results

Table 4 shows the features selected in each of the five executions and their accuracy on the labeled data after each greedy iteration.

iter.	feature	accuracy
1	$f_{\text{hand\_fret}}$	49.22
2	f_{stretch}	56.85
3	$f_{\text{number\_of\_fingers}}$	59.03
4	$f_{\text{hand\_first}}$	59.03

(4.1) Execution 1

iter.	feature	accuracy
1	$f_{\text{hand\_fret}}$	49.22
2	f_{stretch}	57.16
3	$f_{\text{number\_of\_fingers}}$	59.37
4	$f_{\text{hand\_first}}$	59.37

(4.2) Execution 2

iter.	feature	accuracy
1	$f_{\text{hand\_fret}}$	49.22
2	f_{stretch}	56.85
3	$f_{\text{number\_of\_fingers}}$	59.03
4	$f_{\text{hand\_fitts}}$	59.44

(4.3) Execution 3

iter.	feature	accuracy
1	$f_{\text{hand\_fret}}$	49.22
2	f_{stretch}	57.02
3	$f_{\text{number\_of\_fingers}}$	58.85
4	$f_{\text{max\_fingers}}$	58.96
5	$f_{\text{width\_fretwise}}$	58.99
6	$f_{\text{hand\_fitts}}$	59.41

(4.4) Execution 4

iter.	feature	accuracy
1	$f_{\text{hand\_fret}}$	49.43
2	f_{stretch}	56.68
3	$f_{\text{number\_of\_fingers}}$	59.03

(4.5) Execution 5

Table 4.: Results of the five executions of greedy forward selection

Table 5 lists the final selection of features. The feature set consists of a mixture of features taken from the categories of cognitive and biomechanical cost function components (see table 3). Transitional and static features are represented in the selection. A bias towards features concerning the hand can be observed (4 hand features vs. 1 finger feature).

feature	weight λ
$f_{\text{hand\_fret}}$	0.4352
$f_{\text{hand\_fitts}}$	0.1560
f_{stretch}	0.1411
$f_{\text{number\_of\_fingers}}$	0.1167
$f_{\text{hand\_first}}$	0.0316

Table 5.: Final selection of features and their associated weight λ_i after training the CRF

The final CRF was trained on the full pieces (not the split up versions) using the selected feature set and $\sigma^2 = 10$. The weights λ of each feature after training are listed in Table 5. With nearly triple the weight of the next feature, the weighting of $f_{\text{hand\_fret}}$ reflects the importance of choosing fretboard positions in low frets.

7 Results

The topic of this chapter is to evaluate the CRF approach presented in the previous chapters. The approach is evaluated by its accuracy in reproducing known fingerings of guitar pieces and by a qualitative assessment of its generated fingerings.

7.1 Accuracy in Reproducing Known Fingerings

This section covers the accuracy of the CRF in reproducing the fingerings of guitar pieces which were included in previous evaluations by Radicioni et al. [19] and Tuohy et al. [32].

7.1.1 Datasets

Dataset A: Six Guitar Pieces used by Radicioni et al. [19]

This dataset consists of six guitar pieces from the 19th century [19]. Radicioni et al. chose two representative pieces for each class in their categorization of passages (the classes being `CHO`, `MEL`, `MIX` – see section 3.1). Three of the six pieces were included in shortened form with information on the exact bars included for each piece.

An inconsistency was found in the description of the dataset: *Op. 35 No. 3*, written by Fernando Sor, is reported to be an Andantino belonging to the `CHO` class. However, *No. 3* in this opus is not an Andantino but a Larghetto[1)]. The Andantino would have been *No. 2* in the same opus.[2)] It is unclear which one of the two pieces was ultimately included in the dataset as neither of both are particularly chord-like.

The fingerings of the six pieces chosen by Radicioni et al. were completed by an independent guitar expert and were not made available to the public. Except for the completion of the fingerings and the shortened pieces, no information on preprocessing steps such as the treatment of repetitions or trill notes is stated.

The authors were not available to discuss these issues. For this reason, the most likely composition of the dataset and the preprocessing steps involved had to be deduced from hints on the number of notes in comparison to sheet music of the pieces found online. The composition which came closest to the reported number of notes (953 vs. the 948 notes stated by Radicioni et al.) included the Larghetto from Op. 35 No. 3 and removed repetitions and trill notes where applicable. Table 6 shows the deduced composition of the dataset.

In absence of the original dataset, fingerings for the pieces were completed manually by the author of this thesis (who has more than 15 years of experience in playing the classical guitar). The basis for the fingerings was partially fingered sheet music found online. With the exception of one piece, reliable sources for sheet music were found (see appendix B). The manually completed fingerings were not verified by an independent guitarist. However, as Radicioni et al. mention themselves, the six pieces "do not present particular difficulties" which is why "[they] assume that different performers would have provided substantially similar results" [19, p. 20]. It can therefore be assumed that the fingerings evaluated in this thesis are reasonably similar to the ones employed by Radicioni et al. [19]. The MIDI files for the pieces were created to match the exact note values stated in the sheet music. Links to the sheet music with partial fingerings are listed in appendix B.

1) See page 3 of the sheet music at http://carkiv.musikverk.se/www/boije/Boije_0481.pdf (visited on 03/21/2016) from the Boije collection, a collection of free classical guitar sheet music.

2) See page 2 in the same PDF.

Class	Composer	Title
CHO	D. Aguado	Estudio No. 3
	F. Sor	Op. 35 No. 3 Larghetto
MEL	M. Carcassi	Op. 60 No. 7, bars 1–8
	M. Giuliani	Op. 50 No. 13
MIX	F. Carulli	Op. 121 No. 15 Siciliana, bars 1–22
	F. Sor	Op. 21 No. 6, bars 1–16

Table 6.: The pieces of dataset A grouped by their class (adapted from Radicioni et al. [19, table 6])

Dataset B: Three Guitar Pieces used by Radicioni et al. [21]

Three of the six pieces from dataset A were used by Radicioni et al. [21] to evaluate their previous path-based approach for generating fingerings which did not yet support `MIX` passages. The dataset consists of one piece of each class from dataset A, however with one difference: There is no mention of any shortening in [21] which lets one assume that the whole pieces were evaluated. Table 7 lists the composition of this dataset.

Composer	Title
D. Aguado	Estudio No. 3
M. Carcassi	Op. 60 No. 7
F. Carulli	Op. 121 No. 15 Siciliana

Table 7.: The pieces of dataset B (adapted from Radicioni et al. [21])

The dataset was included in this thesis despite the similarity to dataset A because accuracies were reported by Radicioni et al. [21] themselves using their path-based approach as well as by Tuohy et al. [31] using their genetic algorithm. As mentioned in the literature review in section 3.2, Tuohy et al. consider the creation of fingerings for this dataset to be "not particularly difficult" because an optimal fingering can be created by choosing the fretboard position of the lowest fret for most notes [32].

Analogous to their following paper, Radicioni et al. let an independent guitar expert complete the fingerings but refrained from making them available to the public.

The MIDI files and fingerings were therefore created by the same procedure as for dataset A using the same sheet music.

Dataset C: Excerpts Taken from classtab.org used by Tuohy et al. [32]

This test set and the associated research were already mentioned in the literature review in section 3.3.

The dataset used in the evaluation by Tuohy et al. [32] consists of 65 excerpts of guitar tablature (reportedly) taken from the website classtab.org. The dataset and the fingerings generated for each excerpt were uploaded to the authors website[3)]. The excerpts are represented exclusively by ASCII tablature without information on note values or tempo. Because Tuohy et al. only analyzed the assignment of fretboard positions to notes, no finger assignments are present in the dataset. The length of the excerpts varies between two and six bars.

[3)] http://www.ai.uga.edu/tuohy/excerpts.html (dead link) – An archived version from 2007 is available at archive.org: http://web.archive.org/web/20070724175959/http://ai.uga.edu/tuohy/excerpts.html (visited on 03/21/2016)

The quality of the dataset is mediocre at best. Tuohy et al. rightfully state that they "have no reason to believe that [their] data are devoid of [irrelevant inputs and noisy training data]" [32, p. 2], however, major issues could have been easily resolved. The issues found with the dataset are:

- The dataset includes tablature from contemporary artists which are not (and were not at the time) available on classtab.org.
- One excerpt appears twice in the dataset (*Lucy in the Sky with Diamonds* by The Beatles).
- Multiple excerpts were mislabeled, in some cases to the point where the original piece was unidentifiable.
- Several of the fingerings are of low quality and were reportedly created by beginners to tablature creation.

To prepare the dataset for the usage in this thesis, the tablature excerpts were manually transformed into the previously described CSV format. Each excerpt was complemented by a MIDI representation: The MIDI files were taken from classtab.org if available, otherwise matching MIDI files were derived from sheet music found online. In total, 59 of the 65 pieces were prepared for the CRF.

It goes without saying that running the CRF on a dataset enhanced by MIDI files impairs the comparability to the results achieved by Tuohy et al. [32] since the CRF is provided more information on the dependencies between notes. The dataset was nevertheless included in the evaluation because it is the only dataset appearing in published literature which contains more than six pieces.

Characteristics of the Datasets

The remark on the characteristics of dataset B by Tuohy et al. [31] becomes apparent when looking at the occurrence probability of frets in the fingerings, as shown for all datasets in figure 15. All datasets exhibit a considerable amount of notes played on open strings which can be recognized from the peaks at fret 0. The peaks at frets 2, 5, 7 and 12 stem from the fretboard positions of important notes belonging to the keys A and E, in which most guitar pieces are written. Datasets A and B are noticeably skewed towards the lower frets whereas dataset C and the training dataset are more diverse.

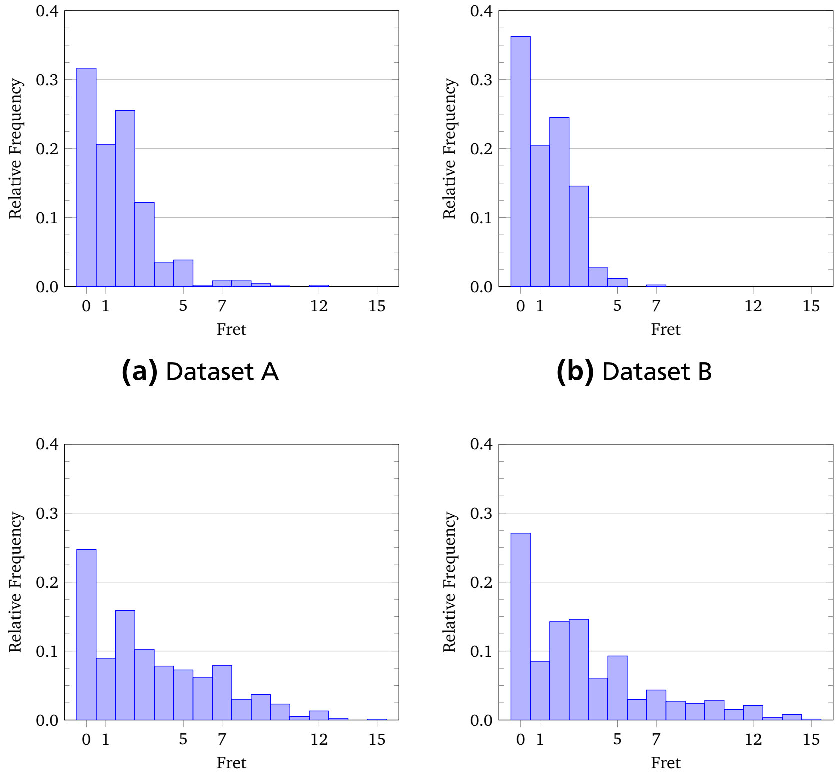
7.1.2 Measures of Accuracy

Two measures of accuracy were employed in accordance with the works of Radicioni et al. and Tuohy et al.:

$\langle s, fr, fi \rangle$ Accuracy: The percentage of matching fingered positions when comparing generated fingerings to reference fingerings note by note (the same accuracy measure employed in the feature selection).

$\langle s, fr \rangle$ Accuracy: A relaxed version of the above ignoring finger assignments, i. e. the percentage of matching fretboard positions when comparing generated fingerings to reference fingerings note by note.

The reasoning behind the second measure is (apart from being able to compare approaches which do not assign fingers to fretboard positions, such as the genetic algorithm by Tuohy et al. [31]) to reduce the influence of personal preference on the computed accuracy, since choice of a finger for a fretboard position is (even) more of subjective nature than the choice of a fretboard position itself. For the reasons previously stated in section 6.2.3, the significance of both measures should not be overinterpretated.



(a) Dataset A **(b)** Dataset B

(c) Dataset C **(d)** Training Dataset

Figure 15.: Relative frequency of frets in the reference fingerings of each dataset

7.1.3 Achieved Accuracy

Table 8 displays the accuracy of the CRF achieved on dataset A in comparison to the reported accuracy of the path-based approach by Radicioni et al. [19]. The accuracy is presented separately for each class of pieces and averaged over all classes (weighted by the number of notes of each class). The CRF outperforms the approach by Radicioni et al. across all classes with respect to finding the reference fretboard positions but falls behind when finger assignments are taken into account.

	$\langle s, fr, fi\rangle$ Accuracy		$\langle s, fr\rangle$ Accuracy	
	Radicioni et al. [19]	CRF	Radicioni et al. [19]	CRF
`CHO`	88.21	72.14	98.40	99.64
`MEL`	87.02	85.11	98.25	100.00
`MIX`	83.31	75.63	95.77	98.49
total	86.18	77.40	97.47	99.27

Table 8.: Accuracy reached on dataset A

Likewise, table 9 shows a comparison of the accuracies achieved on dataset B, including the total accuracy reported by Tuohy et al. [31] using their genetic algorithm. Tuohy et al. did not report detailed results for each piece. Again, the CRF prevails for the second accuracy measure, scoring 100 % accuracy on all three pieces, but deviates more from the reference fingerings once finger assignments are included.

Finally, the performance of the CRF on dataset C is shown in table 10: Despite the additional knowledge offered by the MIDI files, the CRF performs worse than the genetic algorithm. The second measure was not evaluated for this dataset, as it does not include finger assignments.

	$\langle s, fr, fi \rangle$ Accuracy		$\langle s, fr \rangle$ Accuracy		
	Radicioni et al. [21]	CRF	Radicioni et al. [21]	Tuohy et al. [31]	CRF
D. Aguado – Estudio No. 3	95.75	84.44	100.00		100.00
M. Carcassi – Op. 60 No. 7	87.69	82.53	95.86		100.00
F. Carulli – Op. 121 No. 15	88.41	82.76	97.60		100.00
total	90.61	82.82	97.82	98.90	100.00

Table 9.: Accuracy reached on dataset B

	$\langle s, fr \rangle$ Accuracy	
	Tuohy et al. [32] (GA)	CRF
total	86.90	84.48[1)]

[1)] obtained on a reduced dataset with information not available to the GA

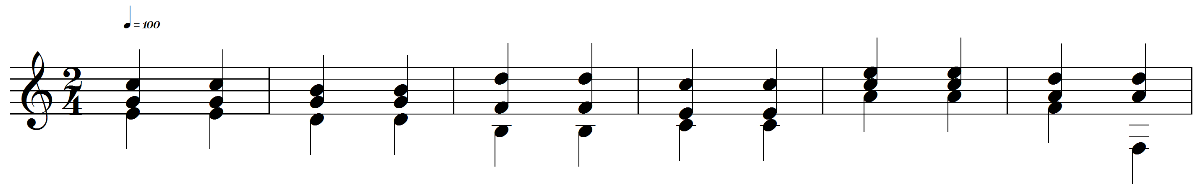
Table 10.: Accuracy reached on dataset C

7.2 Qualitative Assessment from the Viewpoint of a Guitarist

In the following, the quality of four exemplary fingerings is pointed out compared to their reference fingering.

Dionisio Aguado – Estudio No. 3

Estudio No. 3 by Dionisio Aguado is a study on chords appearing in dataset A and B. Figure 16 shows the first six bars of the sheet music and reference fingering as well as the fingering generated by the CRF.



(a) Sheet music of the piece

```
E4|--------|--------|--------|--------|--0--0--|--------|
B3|--1--1--|--0--0--|--3--3--|--1--1--|--1--1--|--3--3--|
G3|--0--0--|--0--0--|--------|--------|--2--2--|--2--2--|
D3|--2--2--|--0--0--|--3--3--|--2--2--|--------|--3-----|
A2|--------|--------|--2--2--|--3--3--|--------|--------|
E2|--------|--------|--------|--------|--------|-----1--|
     1  1              4  4     1  1     1  1     4  4
     2  2              3  3     2  2     2  2     2  2
                       2  2     3  3              3  1
```

(b) Tablature of the reference fingering

```
E4|--------|--------|--------|--------|--0--0--|--------|
B3|--1--1--|--0--0--|--3--3--|--1--1--|--1--1--|--3--3--|
G3|--0--0--|--0--0--|--------|--------|--2--2--|--2--2--|
D3|--2--2--|--0--0--|--3--3--|--2--2--|--------|--3-----|
A2|--------|--------|--2--2--|--3--3--|--------|--------|
E2|--------|--------|--------|--------|--------|-----1--|
     1  1              4  4     1  1     1  1     4  4
     4  4              3  3     2  2     4  4     2  2
                       2  2     4  4              3  1
```

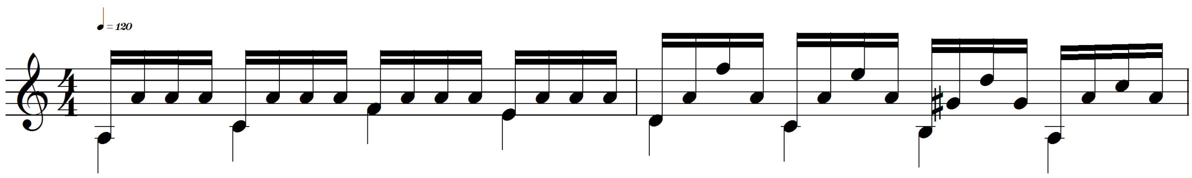
(c) Tablature of the fingering generated by the CRF

Figure 16.: Fingering comparison of the first six bars from *D. Aguado – Estudio No. 3* taken from datasets A / B – deviations from the reference fingering are highlighted

The fingering generated by the CRF matches the reference fingering exactly with respect to the fretboard positions. In bars 1 and 5, the CRF chooses unusual finger assignments and places the index and little finger in neighboring frets, which is a decision guitarists would consider unnecessary and uncomfortable to play. The deviation in finger assignment in bar 4 is a matter of personal preference.

Matteo Carcassi – Op. 60 No. 7

Op. 60 No. 7 by Matteo Carcassi is a melodic piece appearing in dataset A and B. The first two bars of sheet music and fingerings are shown in figure 17.



(a) Sheet music of the piece

```
E4|-------------------------------------------------|-------1-----------0-----------------------------|
B3|-------------------------------------------------|-------------------------------3-----------1-----|
G3|----2--2--2-----2--2--2-----2--2--2-----2--2--2--|----2-----2-----2-----2-----1-----1-----2-----2--|
D3|-------------------------3-----------2-----------|-0-----------------------------------------------|
A2|-0-----------3-----------------------------------|-------------3-----------2-----------0-----------|
E2|-------------------------------------------------|-------------------------------------------------|
       2  2  2  3  2  2  2  4  2  2  2  1  2  2  2       2  1  2  3  2     2  2  1  4  1     3  1  3
```

(b) Tablature of the reference fingering

```
E4|-------------------------------------------------|-------1-----------0-----------------------------|
B3|-------------------------------------------------|-------------------------------3-----------1-----|
G3|----2--2--2-----2--2--2-----2--2--2-----2--2--2--|----2-----2-----2-----2-----1-----1-----2-----2--|
D3|-------------------------3-----------2-----------|-0-----------------------------------------------|
A2|-0-----------3-----------------------------------|-------------3-----------2-----------0-----------|
E2|-------------------------------------------------|-------------------------------------------------|
       2  2  2  3  2  2  2  3  2  2  2  2  4  4  4       2  1  2  3  2     2  2  1  4  1     2  1  2
```

(c) Tablature of the fingering generated by the CRF

Figure 17.: Fingering comparison of the first two bars from *M. Carcassi – Op. 60 No. 7* taken from datasets A / B – deviations from the reference fingering are highlighted

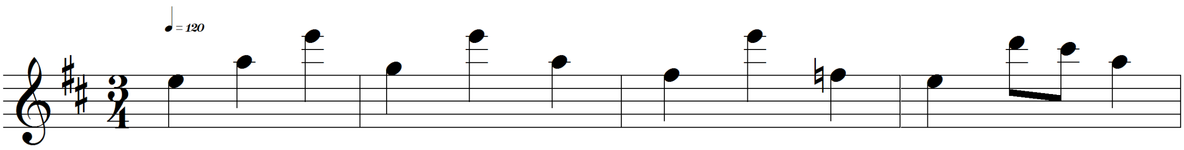
Since the piece is supposed to be played *allegro* (fast), the reference fingering attempts to ease the transitions between forms. Throughout most of the two bars, the middle finger is placed on the third string to play A3 while the other fingers are responsible for the melody. Near the end of the second bar, the idling ring finger takes over the position previously played by the middle finger to avoid having to move the latter into position in time.

As for the previous piece, the generated fingering matches the fretboard positions of the reference fingering while the finger assignments deviate for several notes. The generated fingering requires unnecessary movement of the middle and ring finger in the first bar and misses the opportunity of choosing the ring finger in the second bar.

Excerpt of The Beatles – Lucy in the Sky with Diamonds

Lucy in the Sky with Diamonds is a well-known song by The Beatles of which an excerpt of four bars was included in dataset C. The CRF reached a low accuracy of 38.46 % with respect to fretboard positions compared to the reference tablature. The reference tablature and the tablature generated by Tuohy et al. [32] as well as the fingering generated by the CRF are shown in figure 18.

The reference fingering suggests to play the notes on alternating strings while keeping the hand positioned in the 9th fret. Whereas the genetic algorithm roughly produces the reference fingering, the



(a) Sheet music of the piece

```
E4|---------12--|------12------|------12------|-----10--9------|
B3|-----10------|----------10--|--------------|------------10--|
G3|--9----------|--12----------|--11------10--|--9-------------|
D3|-------------|--------------|--------------|----------------|
A2|-------------|--------------|--------------|----------------|
E2|-------------|--------------|--------------|----------------|
```

(b) Tablature of the reference fingering

```
E4|--0---5--12--|------12------|------12------|-----10--9------|
B3|-------------|---8------10--|---7----------|------------10--|
G3|-------------|--------------|----------10--|--9-------------|
D3|-------------|--------------|--------------|----------------|
A2|-------------|--------------|--------------|----------------|
E2|-------------|--------------|--------------|----------------|
```

(c) Tablature generated by the genetic algorithm of Tuohy et al. [32]

```
E4|--0---5--12--|---3--12---5--|---2--12---1--|--0--10--9---5--|
B3|-------------|--------------|--------------|----------------|
G3|-------------|--------------|--------------|----------------|
D3|-------------|--------------|--------------|----------------|
A2|-------------|--------------|--------------|----------------|
E2|-------------|--------------|--------------|----------------|
         4    4     3    4    4     2    4    1        4   4    4
```

(d) Tablature of the fingering generated by the CRF

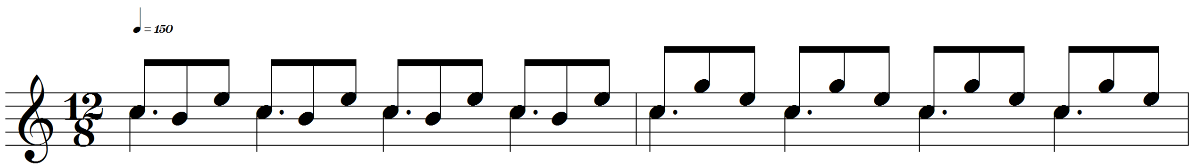
Figure 18.: Fingering comparison of an excerpt from *The Beatles – Lucy in the Sky with Diamonds*, taken from dataset C – deviations from the reference tablature are highlighted[4)]

CRF places all notes on the first string, thereby requiring a lot of hand movement to play the excerpt. Although playing multiple notes on one string while moving the hand along the fretboard is justifiable in some cases, the fingering generated by the CRF is undeniably of poor quality and would be rejected by any guitarist.

4) The discerning reader will notice that both tablature and sheet music are incorrect in the sense that they do not match the actual notes of the song. The tablature was taken from the dataset provided by Tuohy et al. [32] without further modifications. The MIDI was created to match said tablature.

Excerpt of Leo Brouwer – Omaggio a Debussy

Omaggio a Debussy by Leo Brouwer is a melodic piece which is also included in dataset C. 33.33 % of the fretboard positions in the fingering created by the CRF match the positions from the reference tablature. The reference tablature, the tablature generated by Tuohy et al. [32] and fingering created by the CRF can be found in figure 19.



(a) Sheet music of the piece

```
E4|--------0--------0--------0--------0--|--------0--------0--------0--------0--|
B3|-----0--------0--------0--------0-----|-----8--------8--------8--------8-----|
G3|--5--------5--------5--------5--------|--5--------5--------5--------5--------|
D3|--------------------------------------|--------------------------------------|
A2|--------------------------------------|--------------------------------------|
E2|--------------------------------------|--------------------------------------|
```

(b) Tablature of the reference fingering

```
E4|--------0--------0--------0--------0--|-----3--0-----3--0-----3--0-----3--0--|
B3|--1--0-----1--0-----1--0-----1--0-----|--1--------1--------1--------1--------|
G3|--------------------------------------|--------------------------------------|
D3|--------------------------------------|--------------------------------------|
A2|--------------------------------------|--------------------------------------|
E2|--------------------------------------|--------------------------------------|
```

(c) Tablature generated by the genetic algorithm of Tuohy et al. [32]

```
E4|--------0--------0--------0--------0--|-----3--0-----3--0-----3--0-----3--0--|
B3|--1--------1--------1--------1--------|--1--------1--------1--------1--------|
G3|-----4--------4--------4--------4-----|--------------------------------------|
D3|--------------------------------------|--------------------------------------|
A2|--------------------------------------|--------------------------------------|
E2|--------------------------------------|--------------------------------------|
     1  4     1  4     1  4     1  4         1  4     1  4     1  4     1  4
```

(d) Tablature of the fingering generated by the CRF

Figure 19.: Fingering comparison of an excerpt from *L. Brouwer – Omaggio a Debussy*, taken from dataset C – deviations from the reference tablature are highlighted

The reference fingering places the notes on alternating strings in a consistent pattern. A guitarist would take advantage of such a pattern by choosing a right hand fingering strategy where one finger exclusively plays on one string. One option would be to pluck the third string with the thumb, the second with the index finger and the first string with the middle finger.

The fingering generated by the CRF follows a similar pattern at first but breaks with the pattern midway through the excerpt, forcing guitarists to adjust their right hand fingering strategy to a less favorable strategy where the exclusivity between fingers and strings is not given anymore. Contrary to the previous example piece, the generated fingering is easily playable but is nonetheless unsatisfying for guitarists.

Summary

The perceived quality of the generated fingerings is mixed. Apart from occasional bad decisions, the choice of fretboard positions for notes is appropriate. The assignment of fingers to fretboard positions is convincing for cases in which the assignment is dictated by the distances between the fretboard positions, by the number of notes in a form or by held notes. In any other case, the finger assignments appear unsystematic.

The biomechanical playability of the individual forms needs no comment as it is guaranteed by the presented approach.

8 Discussion

The fingering generated for *Op. 60 No. 7* shows weaknesses directly related to the first-order Markov assumption: The chance of using the ring finger to play A3 near the end of the second bar is missed because CRF features of first order only have access to two consecutive forms. In this particular case, the features are given two forms belonging to bass note A2 and the respective note A3 but are unaware of the role of the middle finger in previous forms.

The Markov assumption has a more serious impact on *Lucy in the Sky with Diamonds*: The low quality fingering arises from the Markov assumption in conjunction with the strongly weighted $f_{\text{hand\_fret}}$ feature and the lack of a feature penalizing hand movement in the feature set. After each time playing E5 in the 12th fret on the first string, the CRF cannot look back enough notes to realize that it is beneficial to keep the hand positioned near the corpus. Instead, $f_{\text{hand\_fret}}$ pulls the hand towards the lower frets without penalty for the resulting hand movement. The hand eventually needs to be moved towards the corpus again for playing the next E5 (for which there are no better alternative fretboard positions except for an even higher position in the 17th fret on the second string).

The genetic algorithm produces a better fingering for this piece because it optimizes the whole fingering at once, providing the fitness function with the fretboard position of all fingers in each invocation of its fitness function.

The presented CRF approach was built around the difficulty aspects of guitar play stated in section 2.5, mainly concerning the actions of the left hand. Actions of the right hand were taken out of the equation due to their low impact on difficulty.

Consequently, the CRF creates a fingering for *Omaggio a Debussy* which is easy to play for the left hand but is not sophisticated enough to convince guitarists due to the lacking respect for actions of the right hand. The fingering created by the genetic algorithm exhibits the same flaw.

Apart from this issue, the generated fingering for *Omaggio a Debussy* demonstrates the problem of evaluating a fingering via note by note comparison to a reference fingerings: The first bar of the fingering created by the CRF would be accepted by guitarists but receives a low $\langle s, fr\rangle$ accuracy of 33 % because few fretboard positions coincide with the reference fingering. Overall however, the accuracies computed for the three evaluated datasets reflect the view of a guitarist on the fingerings.

The low $\langle s, fr\, fi\rangle$ accuracy and the observation of unsystematic finger assignments can be traced back to the lack of features in the feature set which break the tie between equally suited fingers for a fretboard position. Additionally, no feature is present which penalizes finger movement. The approach by Radicioni et al. [19] fares better in this respect due to the judgment of comfort spans between finger pairs in its cost function (which are not covered in detail in their paper).

On the other hand, the high $\langle s, fr\rangle$ accuracy on datasets A and B can be explained by the way the number of viable fretboard positions is restricted for each note. Naturally, the fewer fretboard positions are possible for one note, the higher the chance of picking the "correct" position chosen in the reference fingering. Every piece within the datasets A and B exhibits at least one of the following characteristics, benefiting the creation of fingerings with the presented CRF approach:

1. It contains a large amount of chords with many notes, reducing the size of the sets $PLAYABLE\_CONSISTENT_{x,tu}$.
2. It contains notes near the low or high end of the pitch spectrum of a guitar for which few fretboard positions exist.
3. It contains many sections with held notes which restrict the positions of other notes, creating fewer links between layers of the SFSA.

To conclude, the pieces are indeed easier cases to create fingerings for, as claimed by Tuohy et al. [31].

The second trait applies to fewer pieces of dataset C (as reflected in the fret distribution shown in figure 15c), rendering it more difficult to reach high $\langle s, fr \rangle$ accuracies.

To gain a better insight into the performance of the CRF on dataset C, it is helpful to plot the $\langle s, fr \rangle$ accuracy broken down by the frets of the reference fingered positions, as shown in figure 20. As an example, slightly more than 60 % of the notes played in the fifth fret in the reference fingerings were also assigned a position in the fifth fret by the CRF.

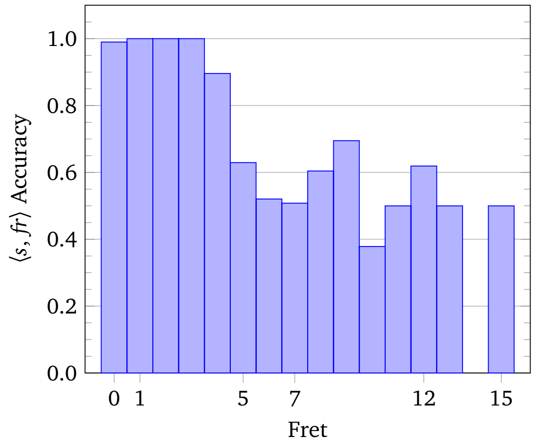


Figure 20.: $\langle s, fr \rangle$ Accuracy on dataset C, broken down by the frets of the reference fingered positions

The plot generally reveals that the accuracy diminishes for higher frets. A 30 % drop in accuracy can be observed between the fourth and the fifth fret.

Remember that due to the overlapping pitch ranges of a guitar, the fretboard positions of one note on different strings are mostly five frets apart from each other (see section 2.1), therefore every note in the reference fingering (except for high notes) played in the fifth fret or above can also be played five frets lower. The drop in accuracy again demonstrates the tendency of the CRF to choose these lower fretboard positions over the higher alternatives, influenced by the $f_{\text{hand\_fret}}$ feature.

Note that despite accuracies of around 50 % for any fret higher than the fifth, the CRF reaches a total $\langle s, fr \rangle$ accuracy of 84.48 % due to the prevalence of lower frets in the dataset (see figure 15c).

8.1 Impact of the Feature Set on the Results

This section analyzes the impact of different several feature sets on the accuracy of the CRF and quality of the generated fingerings.

The feature set obtained by feature selection serves as the baseline. The following feature sets were chosen for the comparison:

Baseline feature set with fretwise distance computation: To assess the influence of incorporating knowledge on the physical dimensions of guitar and guitarist, a feature set is chosen which resembles the baseline feature set but replaces the $f_{\text{hand\_stretch}}$ and $f_{\text{hand\_fitts}}$ features by their counterparts based on fretwise distances.

HMM state transition probabilities: A comparison to the HMM approach pursued by Hori et al. [9] is established by choosing a feature set which consists only of f_{hmm} (see section 5.4.7).

The exact composition of each feature set is listed in table 11.

Table 12 shows the accuracies achieved with each feature set. On datasets A and B, the baseline $\langle s, fr \rangle$ accuracy is matched but not surpassed by the other two feature sets. The HMM feature set fares slightly better on dataset C than the baseline. Regarding the $\langle s, fr, fi \rangle$ accuracy measure, the baseline was

feature	weight λ
$f_{\text{hand\_fret}}$	0.4352
$f_{\text{hand\_fitts}}$	0.1560
f_{stretch}	0.1411
$f_{\text{number\_of\_fingers}}$	0.1167
$f_{\text{hand\_first}}$	0.0316

(11.1) Baseline

feature	weight λ
$f_{\text{hand\_fret}}$	0.3909
$f_{\text{hand\_fitts\_fretwise}}$	0.3238
$f_{\text{stretch\_fretwise}}$	0.1127
$f_{\text{number\_of\_fingers}}$	0.1010
$f_{\text{hand\_first}}$	0.0102

(11.2) Fretwise

feature	weight λ
f_{hmm}	1.0000

(11.3) HMM

Table 11.: Composition of the three feature sets included in the comparison

Dataset		$\langle s, fr, fi \rangle$ Accuracy			$\langle s, fr \rangle$ Accuracy		
		Baseline	Fretwise	HMM	Baseline	Fretwise	HMM
Dataset A	`CHO`	72.14	**75.71**	71.43	99.64	99.29	99.64
	`MEL`	85.11	81.56	78.72	100.00	97.52	99.29
	`MIX`	75.63	**76.13**	**78.89**	98.49	94.73	97.24
	total	77.40	**77.60**	76.66	99.27	96.88	98.54
Dataset B	D. Aguado – Estudio No. 3	84.44	**92.22**	80.00	100.00	100.00	100.00
	M. Carcassi – Op. 60 No. 7	82.53	77.93	80.92	100.00	97.01	98.62
	F. Carulli – Op. 121 No. 15	82.76	81.50	**85.89**	100.00	96.55	100.00
	total	82.82	80.81	82.70	100.00	97.16	99.29
Dataset C	total				84.48	80.16	**85.04**

Table 12.: Accuracies achieved by each feature set on the three datasets – emphasized: accuracies surpassing the accuracy of the baseline feature set

outmatched on two different pieces of dataset B. This is also reflected in the accuracies of dataset A due to the overlap between the two datasets. The largest improvement over the baseline feature set can be observed for *D. Aguado – Estudio No. 3* with nearly 8 % more matches. This difference is not as significant as it might appear because the guitar piece in question consists of only 90 notes, meaning the feature set with fretwise features matches only 7 notes more than the baseline feature set.

Overall, the accuracies are comparable and exhibit no major increase or loss in accuracy. The baseline feature set utilizing physical dimensions shows no significant improvement but has a slight edge over the other feature sets with respect to choosing matching fretboard positions. Regarding the quality of the fingerings, similar observations were made: The generated fingerings of either alternative feature set exhibited no notable improvements or weaknesses compared to the fingerings generated by the baseline feature set.

9 Conclusion and Future Work

This chapter summarizes the approach and the findings of this thesis and provides suggestions for further research on the topic.

9.1 Conclusion

The topic of this master's thesis was to extend the capabilities of approaches to the guitar fingering which are based on finding an optimal path in a graph (path-based approaches).

The aspired extensions were to lift the restriction of path-based approaches being only applicable for guitar pieces of a special structure (such as melodies or chord sequences) and to investigate the potential of incorporating physical properties of instrument and performer into the cost function.

A linear-chain conditional random field (CRF) was chosen as the underlying model because of its beneficial properties for the guitar fingering problem and the above extensions.

The support for guitar pieces of arbitrary structure was realized by automatically dividing a given piece into segments in which all notes remain constant (*constant segments*). These segments consist of a chord, a duration, their time of occurrence in the piece and additional information necessary to identify notes held across multiple segments.

For each chord inside the segments of a piece, a set of viable chord fingerings (a set of *forms*) was generated. To this end, a technique described by Radicioni et al. [21] was adopted. It involves the formulation of a constraint satisfaction problem whose constraints model principles of guitar play and biomechanical limitations of the human hand. Given a chord, the set of solutions to such a constraint satisfaction problem represent the set of viable forms.

These sets of forms were organized in a layered graph in which each layer contains the viable forms of one constant segment. The concept of layered graphs was necessary to facilitate the use of CRFs for the guitar fingering problem. To generate a fingering for a guitar piece, the CRF is provided with the sequence of constant segments and the layered graph derived from the piece.

Several CRF features (components of the cost function) were defined which judge the biomechanical and cognitive difficulty of guitar playing. Besides features adopted from the cost functions of other authors, new features incorporating physical properties were defined such as a penalty for forms which require stretching of the left hand based on the true distances found on a guitar.

The subset of features with the highest explanatory power was determined by feature selection. The CRF evaluated throughout the rest of the thesis was trained on 10 guitar pieces with known fingerings using the feature subset just mentioned.

The CRF approach was evaluated on three datasets of guitar pieces with known fingerings which were previously used in evaluations by Radicioni et al. [19] and Tuohy et al. [32]. From the standpoint of a guitarist, the fingerings generated by the CRF were mostly appropriate with respect to the fretboard positions chosen for each note. The quality of the finger assignments was considered to be unsatisfying.

This assessment was reflected in the quantitative comparison between fingerings generated by the CRF and reference fingerings from the three datasets: The fretboard positions of the generated fingerings matched the positions of the reference fingerings for 99.27 %, 100.00 % and 84.48 % of all notes in the respective datasets. Once finger assignments were included in the accuracy measure, the accuracies dropped to 77.40 % and 82.82 % on the first two datasets (finger assignments were not included in the third dataset). Due to one strongly weighted feature which preferred positions in lower frets over positions in higher frets, the CRF reached the highest accuracies for pieces which were played in lower frets.

The mixed quality of the generated fingerings was traced back to the first-order Markov assumption inherent to the presented CRF approach and other path-based approaches. While the assumption guarantees efficient inference, it provides CRF features only with a narrow window of two forms at a time, which hinders their explanatory power.

Experiments with alternative feature sets revealed that features incorporating physical attributes of guitar and guitarist led to a slight increase in accuracy without a noticeable improvement in fingering quality.

To conclude, the presented CRF approach to the guitar fingering problem can be applied to generate fingerings for any given guitar piece. While it has the potential to create convincing fingerings, it is held back by modeling only first-order dependencies between forms in its present state.

9.2 Future Work

The presented approach offers many possibilities for experimentation besides the obvious expansion to higher-order dependencies. Several possibilities are listed in the following section. In the subsequent section, suggestions concerning all approaches to the guitar fingering problem are stated.

9.2.1 Future Work Concerning the CRF Approach

Raise the order of path-based approaches

So far, any of the published path-based approaches to the guitar fingering problem employs a first-order inference technique. It was originally planned to investigate higher-order CRFs for the guitar fingering problem. The idea was dropped to keep the scope of the thesis manageable.

The results of this thesis strongly suggest that raising the order leads an increase in fingering quality. The presented CRF approach is likely to profit the most from raising the order among the family of path-based approaches because its features can access the whole observation sequence: Given the order k, CRF features would have access to the history of forms $y_j, y_{j-1}, \ldots, y_{j-k}$ and their corresponding constant segments from $\vec{x}$. This would allow one to define transitional features incorporating much more knowledge on finger positions and timing than is available in the first-order case.

Include movement of the right hand in the features / the cost function

Neither of the existing approaches to the guitar fingering problem includes movements of the right hand into their cost function. Its inclusion might improve the quality of generated fingerings by tipping the scale in favor of playing on as many strings as possible (for arpeggio sections) or on as few strings as possible (for tremolo sections) for fingerings which would otherwise be considered equally appropriate. The findings of a paper covering the automatic generation of right hand fingering for guitar pieces [28] could be adapted for this purpose.

Integrate knowledge of chord databases

Online chord databases collect "chords" (strictly speaking: forms of chords) which are often used for guitar accompaniment. In order to reduce the effort of memorizing unique forms for guitar pieces, one could favor common forms for chords by testing whether a form appears in such a chord database. The feature was not implemented due to a lack of time. Its utility could be evaluated in future research.

Investigate CRFs for the task of arranging music

Similar to the work of Hori et al. [9, 10] where a HMM is formulated to arrange pieces for the guitar, one could attempt to formulate features which add this ability to a CRF. However, one would need to put more thought into finding the optimal weights since conventional training is likely to be unrealizable due to very sparse labeled data for this purpose.

9.2.2 Future Work Concerning all Approaches

Compile a reliable dataset of guitar pieces and fingerings

This thesis has motivated the need for a dataset of guitar pieces and reliable, high quality fingerings. An optimal dataset would include pieces of a wide range of difficulty levels, tempos, playing styles (barre chords, tremolos, embellishments, etc.) and genres. The dataset could then be used as a common test set across future publications to improve the comparability of individual approaches.

Experiment with other methods to evaluate the quality of fingerings

The percentage of matching fingered positions is commonly used to evaluate the quality of fingerings despite its unrealiability, as pointed out initially by Tuohy et al. [33] and multiple times throughout this thesis. On the other hand, conducting a study with expert guitarists promises reliable results but is a very time consuming process.

An alternative could be to offload the task of evaluating fingerings to online communities of guitar players via crowdsourcing which, given enough participants of different levels of expertise, should yield reliable results with comparably low effort.

Bibliography

[1] Terry Allen and Camille Goudeseune, "Topological Considerations for Tuning and Fingering Stringed Instruments", in: *arXiv preprint arXiv:1105.1383* (2011).

[2] The International MIDI Association, *Standard MIDI-File Format Spec. 1.1*, URL: http://www.music.mcgill.ca/~ich/classes/mumt306/midiformat.pdf (visited on 03/04/2016).

[3] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006, ISBN: 0387310738.

[4] Kalyanmoy Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, New York, NY, USA: John Wiley & Sons, Inc., 2001, ISBN: 047187339X.

[5] Heiko Drewes, "Only One Fitts' Law Formula Please!", in: *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '10, Atlanta, Georgia, USA: ACM, 2010, pp. 2813–2822, ISBN: 978-1-60558-930-5.

[6] Paul M. Fitts, "The information capacity of the human motor system in controlling the amplitude of movement", in: *Journal of experimental psychology* 47.6 (1954), p. 381.

[7] Hank Heijink and Ruud GJ Meulenbroek, "On the complexity of classical guitar playing: functional adaptations to task constraints", in: *Journal of motor behavior* 34.4 (2002), pp. 339–351.

[8] Errol R. Hoffmann, "Which Version/Variation of Fitts' Law? A Critique of Information-Theory Models", in: *Journal of Motor Behavior* 45.3 (2013), pp. 205–215.

[9] Gen Hori, Hirokazu Kameoka, and Shigeki Sagayama, "Input-output HMM applied to automatic arrangement for guitars", in: *Information and Media Technologies* 8.2 (2013), pp. 477–484.

[10] Gen Hori and Shigeki Sagayama, "HMM-based automatic arrangement for guitars with transposition and its implementation", in: *Proceedings of the ICMC | SMC | 2014*, 2014, pp. 1257–1262.

[11] Jiun-Long Huang, Shih-Chuan Chiu, and Man-Kwan Shan, "Towards an automatic music arrangement framework using score reduction", in: *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 8.1 (2012), p. 8.

[12] Roman Klinger and Katrin Tomanek, *Classical Probabilistic Models and Conditional Random Fields*, TU Dortmund, Algorithm Engineering, 2007.

[13] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", in: *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 282–289, ISBN: 1-55860-778-1.

[14] R.M. Mottola, *Calculating Fret Positions*, 2016, URL: http://www.liutaiomottola.com/formulae/fret.htm (visited on 02/29/2016).

[15] Michael O. Rabin, "Probabilistic automata", in: *Information and Control* 6.3 (1963), pp. 230–245, ISSN: 0019-9958.

[16] Oracle, *Java Platform Standard Edition 8 Documentation*, 2016, URL: https://docs.oracle.com/javase/8/docs/ (visited on 03/04/2016).

[17] Charles Prud'homme, Jean-Guillaume Fages, and Xavier Lorca, *Choco Documentation*, TASC, INRIA Rennes, LINA CNRS UMR 6241, COSLING S.A.S., 2015.

[18] Lawrence R Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition", in: *Proceedings of the IEEE* 77.2 (1989), pp. 257–286.

[19] Daniele P. Radicioni and Vincenzo Lombardo, “A Constraint-based Approach for Annotating Music Scores with Gestural Information”, in: *Constraints* 12.4 (2007), pp. 405–428, ISSN: 1572-9354.

[20] Daniele P. Radicioni and Vincenzo Lombardo, “Computational Model of Chord Fingering”, in: *Proceedings of the 27th Annual Conference of the Cognitive Science Society*, 2005, pp. 1791–1796.

[21] Daniele P. Radicioni and Vincenzo Lombardo, “Guitar Fingering for Music Performance”, in: *Proceedings of the International Computer Music Conference (ICMC05)*, 2005, pp. 527–530.

[22] Daniele Radicioni, Luca Anselma, and Vincenzo Lombardo, “A segmentation-based prototype to compute string instruments fingering”, in: *Proceedings of the Conference on Interdisciplinary Musicology*, vol. 17, 2004, p. 97.

[23] Aleksander Radisavljevic and Peter Driessen, “Path difference learning for guitar fingering problem”, in: *Proceedings of the International Computer Music Conference*, vol. 28, 2004.

[24] Nicholas Trevor Rutherford, “FINGAR, a genetic algorithm approach to producing playable guitar tablature with fingering instructions”, in: *Undergraduate project dissertation, Dept. of Computer Sci., Univ. of Sheffield* 15 (2009).

[25] Samir I Sayegh, “Fingering for string instruments with the optimum path paradigm”, in: *Computer Music Journal* (1989), pp. 76–84.

[26] Asher Stern, *CRF*, 2014, URL: https://github.com/asher-stern/CRF (visited on 03/04/2016).

[27] Heinz Teuchert, *Die neue Gitarrenschule - Band 1*, German, Ricordi, 1986.

[28] Ernesto Trajano, Márcio Dahia, et al., “Automatic discovery of right hand fingering in guitar accompaniment”, in: *In Proceedings of the International Computer Music Conference (ICMC’04)*, 2004, pp. 722–725.

[29] Edward Tsang, *Foundations of Constraint Satisfaction*, Academic Press, 1993.

[30] Daniel R Tuohy and Walter D Potter, “GA-based music arranging for guitar”, in: *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, 2006, pp. 1065–1070.

[31] Daniel R Tuohy and Walter D Potter, “Generating guitar tablature with LHF notation via DGA and ANN”, in: *Advances in Applied Artificial Intelligence*, Springer, 2006, pp. 244–253.

[32] Daniel R Tuohy, WD Potter, and Artificial Intelligence Center, “An evolved neural network/HC hybrid for tablature creation in GA-based guitar arranging”, in: *Proceedings of the International Computer Music Conference (ICMC’06)*, 2006.

[33] DR Tuohy and WD Potter, “A genetic algorithm for the automatic generation of playable guitar tablature”, in: *Proceedings of the International Computer Music Conference*, 2005, pp. 499–502.

[34] CH Wagner, “The pianist’s hand: anthropometry and biomechanics”, in: *Ergonomics* 31.1 (1988), pp. 97–131.

[35] Ian H. Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005, ISBN: 0120884070.

A Training Data Composition

Table A.1 lists the pieces chosen from http://classtab.org to conduct feature selection and to train the presented CRF approach. The remarks in table A.2 concern the state of the MIDI files and tablatures as of March 21st, 2016.

ID	Composer and Title	URL (classtab.org/...)	Included Bars		BPM (♩)	# Notes
			Training	Selection		
1	D. Aguado – Nuevo Método para Guitarra – Part 2, Section 1, Chapter 1, Lesson 10	http://www.classtab.org/aguado_nuevo_metodo_para_guitarra_p2_s1_c1_l10.txt	[1–27]	[1–27]	109	90
2	D. Aguado – Nuevo Método para Guitarra – Part 2, Section 2, Chapter 1, Exercise 2	http://www.classtab.org/aguado_nuevo_metodo_para_guitarra_p2_s2_c1_ex02.txt	[1–24]	[1–24]	80	186
3	S. Assad – Farewell	http://www.classtab.org/assad_sergio_farewell.txt	[1–86]	[1–48], [49–86]	100	479
4	A. Barrios – Las Abejas	http://www.classtab.org/barrios_las_abejas.txt	[1–18]	[1–18]	171	378
5	A. Barrios – La Catedral (first movement)	http://www.classtab.org/barrios_la_catedral_1_prelude.txt	[1–49]	[1–24], [25–49]	60	196
6	A. Barrios – Preludio in Cm	http://www.classtab.org/barrios_preludio_in_cm.txt	[1–32]	[1–16], [17–32]	60	379
7	F. Carulli – Op. 114 No. 10	http://www.classtab.org/carulli_op114_no10_prelude_in_em.txt	[1–28]	[1–14], [15–28]	150	330
8	F. Carulli – Op. 114 No. 16	http://www.classtab.org/carulli_op114_no16_prelude_in_a.txt	[1–32]	[1–16], [17–32]	120	460
9	N. Coste – Op. 38 No. 1	http://www.classtab.org/coste_op38_no01_allegretto_in_am.txt	[1–44]	[1–44]	109	295
10	J. Dowland – Orlando sleepeth	http://www.classtab.org/dowland_john_orlando_sleepeth.txt	[1–12]	[1–12]	109	104
						= 2897

Table A.1.: Composition of the training dataset

ID	Remarks
2	• bar 23: wrong MIDI – changed A2 to E2
3	• bar 1 ff.: bad tab – changed ⟨4, 7, 4⟩ to ⟨4, 7, 3⟩ and ⟨3, 5, 3⟩ to ⟨3, 5, 1⟩ • bar 10: wrong MIDI – changed A3 to G3 • bar 38: finger_order constraint violated in tab → B3 at ⟨3,4,4⟩ shortened and D4 at ⟨2,3,2⟩ changed to ⟨2,3,4⟩ • bar 48: bad MIDI – A2 changed to dotted quarter note, otherwise ⟨4, 7, 1⟩ and ⟨4, 10, 4⟩ clash • bars 67/68: wrong tab – changed F2 at ⟨6, 1, 1⟩ to A2 at ⟨5, 0, 0⟩ • bars 79/80: wrong MIDI – A2 should be held across both bars
5	• first fingering chosen • bar 6: wrong MIDI – changed D5 to F♯5 • bar 18: wrong MIDI – changed E3 to F3 • bar 27: wrong MIDI – two F♯4 changed to E4 • bar 28: wrong MIDI – two F♯3 changed to G3 • bar 29: wrong MIDI – changed A3 to A♯3 • bar 35: wrong MIDI – changed D3 to C♯3 • bars 47/48: removed all flageolett notes
6	• first fingering chosen • bar 13: bad tab – changed G at ⟨6, 3, 4⟩ to ⟨6, 3, 3⟩ • bar 21: contains little finger barre chord which can be interpreted as two single positions • bar 31: bad tab – much easier when played in the first fret
8	• first MIDI and first fingering chosen • bar 3: bad tab – changed occurrences of ⟨2, 9, 4⟩ to ⟨2, 9, 3⟩ • bar 6: bad tab – changed to index finger barre chord and middle finger • bar 8: bad tab – changed to index finger barre chord and middle finger • bar 25: bad tab – changed ⟨2, 7, 4⟩ to ⟨2, 7, 3⟩ • bar 37: bad tab – changed ⟨2, 2, 1⟩ to ⟨2, 2, 4⟩
9	• bar 5: bad MIDI – shortened first C to eighth note to avoid violation of finger_order constraint • bar 14: bad MIDI – shortened first chord to eighth notes, is unplayable otherwise • bar 22: bad tab – swapped fingers 2 and 4 in first chord • bar 23: bad MIDI – shortened first chord to eighth notes, is unplayable otherwise • bar 35: bad MIDI – shortened first chord to eighth notes, is unplayable otherwise
10	• all repetitions removed • last bar: bad tab – changed ⟨4, 2, 3⟩ to ⟨4, 2, 2⟩ and ⟨3, 2, 4⟩ to ⟨3, 2, 3⟩

Table A.2.: Remarks and modifications regarding the MIDI files and tablature of the training dataset

B Test Data Composition

Table B.1 lists the sources of the sheet music chosen for the pieces of dataset A and B.

ID	Composer and Title	URL	BPM (♩)	Dataset A		Dataset B	
				Bars	# Notes	Bars	# Notes
1	D. Aguado – Estudio No. 3	http://johnny-jerome.fr/partitions/1800/Aguado_Dionisio/Estudio%20Nr%203%20%28Aguado-Bierschenk%29.pdf	100	[1–16]	90	[1–16]	90
2	M. Carcassi – Op. 60 No. 7	http://carkiv.musikverk.se/www/boije/Boije_0094.pdf	120	[1–8]	121	[1–28]	435
3	F. Carulli – Op. 121 No. 15 Siciliana	http://classicalguitarschool.net/en/Download.aspx?id=1009	75	[1–22]	182	[1–39]	319
4	F. Sor – Op. 35 No. 3 Larghetto	http://classicalguitarschool.net/en/Download.aspx?id=1023	66	[1–26]	186		
5	M. Giuliani – Op. 50 No. 13	http://carkiv.musikverk.se/www/boije/Boije_0139.pdf	109	[1-11]	161		
6	F. Sor – Op. 21 No. 6	http://classicalguitarschool.net/en/Download.aspx?id=1104	75	[1–16]	213		

Table B.1.: Composition of the datasets A and B