University of Music Karlsruhe
Institute for Musicology and Music Informatics

Master Thesis

Automated musical style analysis

Computational exploration of the bass guitar play of Jaco Pastorius on symbolic level

Author:
Frank Zalkow B.A.

Supervisors:
Prof. Dr. Denis Lorrain
Benjamin Graf MSc

Project initiation:
Stephan Brand Wi.-Ing.

September 9, 2015

I would like to thank all who supported this thesis. First and foremost I would like to thank SAP SE who generously supported my work in financial terms, by a concentrated and affectionate working environment as well as with personnel support: Stephan Brand who initiated the project and debated with me about the project from the perspective of a bassist, Pastorius lover and software manager; Benjamin Graf who was available weekly for discussions especially about data science issues. I also want to thank Prof. Denis Lorrain whose encouraging and far-sighted arguments have been inspiring for me as well as Sean Malone, who contributed by occasional mail contact with beneficial comments from his view as bassist, musicologist and Pastorius expert. I also want to thank Julia Blank, Holger Ballweg and Stefan König for proof-reading the complete manuscript.

I hereby certify that this master thesis has been composed by myself, and describes my own work, unless otherwise acknowledged in the text. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged. It has not been accepted in any previous application for a degree. All mentioned web addresses have been accessed in August 2015.

Karlsruhe, September 9, 2015

Contents

1 Introduction

1.1 Preliminary discussion

This thesis concerns computational musical style analysis and introduces a novel approach of style classification and style modification on a symbolic level, tailored to monophonic bass guitar music in general and to the music of Jaco Pastorius specifically. Computationally analyzing musical style has several possible cases of application: Finding an artist in big music collections is an interesting retrieval task, modifying music could have use cases in entertainment industry and in artistic production and giving feedback concerning style could also find usage in music education.

Jaco Pastorius is attributed as one of the pioneers of electric bass guitar and one of the most virtuoso bass players in jazz rock. He is especially well known for his eclectic and melodic playing on the fretless bass. His work in the band *Weather Report*, whereby he introduced himself as "the greatest electric bass player in the world", as well as his solo projects form important cornerstones in the history of jazz rock.[1] Since the fretless electric bass guitar sound is connoted with Pastorius, it is especially challenging, yet promising, to ignore all aspects of "sound" and concentrate only on the symbolic note level. Although one could be tempted to focus on the more superficial sound features in this context, it should be borne in mind that Pastorius was a stunning improviser, who especially made his artistic expression recognizable by the melodic lines he formed in his solos. Following these tracks by computational means will be the guidance of this work. One could criticize that the boundary between composition and interpretation is blurred within this thesis, but the "style" of Pastorius, in the sense considered here, only emerges in the final realization of the music, when composition and interpretation melts together. Modelling his complex style of improvisation is a challenge in contrast to simple walking bass style.[2]

---

1 For an introduction to Jaco Pastorius in general see Bill Milkowski: *Jaco: The Extraordinary and Tragic Life of Jaco Pastorius, "the World's Greatest Bass Player"*, Milwaukee 1995. For a more scholarly appreciation see Mark S. Frandsen: *Forecasting fusion at low frequencies: The bass players of "Weather Report"*, PhD thesis, Texas Tech University, 2010, URL: `http://repositories.tdl.org/ttu-ir/handle/2346/45478`, pp. 70 et seqq. and the references mentioned there.

2 For such simple styles, rules-based algorithms can be sufficient, e.g. Rui Dias/Carlos Guedes: *A Contour-based Jazz Walking Bass Generator*, in: *Proceedings of the Sound and Music Computing Conference* 2013, pp. 305–308.

Musical style recognition is a hard task, even for a human.[3] Several things can be meant by "musical style": Charles Rosen distinguishes between period style and group style in his investigation about the style of Haydn, Mozart and Beethoven.[4] Even although it is a completely different musical context, his findings are interesting to notice: Style is the handling of a musical language and thus is able to express a variety of things. The personal style of an artist can only be described when it's contrasted by a period style. In the case of this thesis one could analogize: It's only meaningful to talk about Pastorius' style when contrasted by other fusion bass guitarists. This can be regarded as close to what is described in the upcoming sections: computationally we try to understand Pastorius' style, not only by his music, but also by counterexamples. The task to define the term *musical style* precisely is a hard one, not attempted here. Even the musicologist Guido Adler states in his early, grand monograph about musical style:

> So [regarding the definition of style] one has to content oneself with periphrases. Style is the center of artistic approaching and conceiving, it proves itself, as Goethe says, as a source of knowledge about deep truth of life, rather than mere sensory observation and replication. [5]

Beside the notion of the impossibility of a precise definition of style, it is hard to take the hint, that style isn't just mere sensory observation, to heart when attempting to model musical style computationally. In the view of Guido Adler a thesis like this may be foredoomed to fail. Nearly 80 years after that, the composer and music philosopher Leonard B. Meyer expresses a rather opposite view:

> Once a musical style has become part of the habit responses of composers, performers, and practiced listeners it may be regarded as a complex system of probabilities. That musical styles are internalized probability

[3] For a study supporting this, see Merilyn Jones: *An Investigation of Skills in Recognition of Musical Style Among College Freshman Music Majors*, in: *Contributions to Music Education* 16 (1989), pp. 77–86.

[4] Charles Rosen: *The Classical Style: Haydn, Mozart, Beethoven*, New York 1995.

[5] Guido Adler: *Der Stil in der Musik. 1. Buch: Prinzipien und Arten des Musikalischen Stils*, Leipzig 1911, p. 5, translation by the author. Original version:

> So muß man sich mit Umschreibungen begnügen. Der Stil ist das Zentrum künstlerischer Behandlung und Erfassung, er erweist sich, wie Goethe sagt, als eine Erkenntnisquelle von viel tieferer Lebenswahrheit, als die bloße sinnliche Beobachtung und Nachbildung.

> systems is demonstrated by the rules of musical grammar and syntax found in textbooks on harmony, counterpoint, and theory in general. [...] For example, we are told that in the tonal harmonic system of Western music the tonic chord is most often followed by the dominant, frequently by the subdominant, sometimes by the submediant, and so forth.[6]

There are a couple of things to notice here: First, Meyer also supports the view that style isn't in the music *per se*, but only when regarded in relation with other systems. Second, style is seen probabilistic, supporting the attempt of this thesis to model style computationally. Third, in the textbooks he mentions, probabilities are used in a very broad sense. Words like *frequently* or *sometimes* aren't enough for the models of this thesis. So one cannot rely on textbooks and has to work through real data.

One of the most prominent researchers, engaging computationally with musical style, especially in symbolic style synthesis, is David Cope.[7] In his basic form his style replication program EMI ("Experiments in Musical Intelligence") has to be fed by over a thousand of user input questions.[8] Cope also attempts to overcome this by automatically analyzing a corpus of music. Roughly, this involves finding what Cope calls signatures, frequently reoccurring sequences, assigning functional units to them[9] and recombining the corpus with special regards to those functional signatures. This leads to impressive results for music with rather homogeneous texture, but I expect it to perform poorly for more eclectic and erratic styles, like the one of Pastorius. Whereas the signatures to be found in the music of Mozart are very evident[10], it is not very plausible to talk about Pastorius' signatures. See figure 50 in section 4 on p. 64 for what could be considered a signature. By far it isn't as earmarking as the Mozart examples of Cope.

Although there stand some arguments against the application of Cope's methods to Pastorius, it still could be an interesting undertaking, since Cope developed his methodology sophisticatedly, far exceeding the rough and basic ideas touched here. But nevertheless, it would involve a considerable amount of work to re-create

6 Leonard B. Meyer: *Meaning in Music and Information Theory*, in: *The Journal of Aesthetics and Art Criticism* 15.4 (1957), pp. 412–424, quotation from p. 414.

7 David Cope: *Computers and Musical Style*, Oxford 1991 (henceforth cited as *Musical Style*), see there pp. 27 et seqq. for a nice short overview about musical style in general, that is also valid within the scope of this thesis; idem: *Virtual Music. Computer Synthesis of Musical Style*, Cambridge 2001 (henceforth cited as *Virtual Music*); idem: *Computer Models of Musical Creativity*, Cambridge 2005 (henceforth cited as *Musical Creativity*).

8 *Musical Style*, pp. 89 et seqq.

9 What Cope calls SPEAC: Statement, Preparation, Extension, Antecedent, Consequent.

10 *Virtual Music*, pp. 109 et seqq.

and implement Cope's methods. Developing an original methodology seemed more research-intensive and thus promising for this thesis.

Cope describes basic categories into which music composing programs fall:

> The approaches [...] include rules-based algorithms, data-driven-programming, genetic algorithms, neural networks, fuzzy logic, mathematical modeling, and sonification. Although there are other ways to program computers to compose music, these seven basic processes represent the most commonly used types.[11]

If one would like to force the approach of this thesis, to fall into these categories, rules-based programming – since in Cope's terminology Markovian processes fall into this category – and data-driven-programming would fit, but a considerable amount of this work wouldn't be described. Especially Cope's category of *genetic algorithms* is too specific and could be generalized to *metaheuristics*, which then would also fit for this thesis. But the exact details about it will become more clear in the upcoming chapter.

Markov chains have a great tradition in music. They found application very early in both computer aided music generation[12] and in musicological studies.[13] More recently, researchers from the Sony Computer Science Laboratory rediscovered Markov chains by combining them with constraint based programming, yielding very interesting results.[14]

Having mentioned those two major branches of automatic music generation, the author recommends a more complete survey of Jose D. Fernández and Francisco Vico[15] for those interested in more branches of this field.

11 *Musical Creativity*, p. 57.

12 Frederick P. Brooks et al.: *An experiment in musical composition*, in: *IRE Transactions on Electronic Computers*, vol. 6, 1957, pp. 175–82; Lejaren Hiller/Leonard Isaacson: *Experimental Music. Composition With an Electronic Computer*, New York 1959; Iannis Xenakis: *Formalized Music. Thought and Mathematics in Composition*, ed. by Sharon Kanach, revised edition, Stuyvesant 1992, pp. 43 et seqq.

13 R. C. Pinkerton: *Information theory and melody*, in: *Scientific American* 194.2 (1956), pp. 77–86; Joseph E. Youngblood: *Style as Information*, in: *Journal of Music Theory*, vol. 2, 1958, pp. 24–35.

14 François Pachet/Pierre Roy: *Markov constraints: steerable generation of Markov sequences*, in: *Constraints* 16.2 (2011), pp. 148–172 (henceforth cited as *Markov Constraints*); François Pachet/Pierre Roy/Gabriele Barbieri: *Finite-length Markov Processes with Constraints*, in: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011, pp. 635–642.

15 Jose D. Fernández/Francisco Vico: *AI Methods in Algorithmic Composition: A Comprehensive Survey*, in: *Journal of Artificial Intelligence Research*, vol. 13, 2013, pp. 513–582.

Research in classifying artists on symbolic level is quite rare. The large majority of research focuses on audio.[16], e.g. evidenced by MIREX (Music Information Retrieval Evaluation eXchange), the main community-based framework for the formal evaluation of Music Information Retrieval algorithms, where all classification tasks are related to audio.[17] When classifying symbolic music, it is rather common to focus on composers,[18] genre[19] or on musical qualities (like "frantic" or "lyrical")[20]. There is a very current project called *Jazzomat*, that is highly related to the present thesis.[21] Although there is some astonishing overlap,[22] the author discovered this project not until a late stage of the present thesis. However, while the *Jazzomat*-project should also be an excellent environment for artist identification, the classification results published so far, are rather concerned with genre.[23] Nevertheless, the fact that there does exist such a related project, gives evidence that this topic is a highly relevant and interesting one.

The next section in this introduction gives an overview about the project of this thesis. Afterwards three chapters from the main part: firstly a novel approach of classifying the music of Pastorius is presented, secondly a novel approach of

---

[16] E.g. Jakob Abesser: *Automatic Transcription of Bass Guitar Tracks applied for Music Genre Classification and Sound Synthesis*, PhD thesis, Technische Universität Ilmenau, 2014, URL: `http://www.db-thueringen.de/servlets/DocumentServlet?id=24846`. This thesis specifically concerns the bass guitar and could be valuable when trying to expand the ideas of the present thesis to audio.

[17] J. Stephen Downie: *The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research*, in: *Acoustical Science and Technology*, vol. 29, 2008, pp. 247–255. For more up-to-date information see the MIREX Wiki: `http://www.music-ir.org/mirex/wiki/MIREX_HOME`. Symbolic Melodic Similarity could be attempted by clustering, but still it isn't classification.

[18] E.g. see Peter van Kranenburg: *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*, PhD thesis, Utrecht University, 2010, URL: `http://dspace.library.uu.nl/handle/1874/179892`, especially chapter 5 (pp. 71–88).

[19] Alexios Kotsifakos et al.: *Genre classification of symbolic music with SMBGT*, in: Fillia Makedon et al. (eds.): *Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments*, 2013, 44:1–44:7.

[20] Roger B. Dannenberg/Belinda Thom/David Watson: *A Machine Learning Approach to Musical Style Recognition*, in: *Proceedings of the International Computer Music Conference*, 1997, pp. 344–347.

[21] Martin Pfleiderer/Klaus Frieler: *The Jazzomat project. Issues and methods for the automatic analysis of jazz improvisations*, in: *Concepts, Experiments, and Fieldwork: Studies in Systematic Musicology and Ethnomusicology* 2010, pp. 279–295; Klaus Frieler et al.: *Introducing the Jazzomat Project and the Melo(S)py Library*, in: *Proceedings of the Third International Workshop on Folk Music Analysis* 2013, pp. 76–78.

[22] In particular focus on jazz solos and feature extraction from symbolic music representations.

[23] Arndt Eppler et al.: *Automatic Style Classification of Jazz Records with Respect to Rhythm, Tempo, and Tonality*, in: Timour Klouche/Eduardo Miranda (eds.): *Proceedings of the 8th Conference on Interdisciplinary Musicology (CIM14)*, 2014.

modifying existing music with the aim of making it closer to the style of Pastorius is introduced and lastly some additional research findings are mentioned. A conclusion, including suggestions for further research, finishes this thesis.

1.2 Overview

This section gives a brief overview about the project content. All steps are described in more detail and with references in the following chapter.

Two main ideas are dominating: The first one is classification. Given a symbolic representation of bass guitar music, the program should decide, whether it is the style of Pastorius or not. The second idea is the one of style modification. The ability of classification should be used for modifying existing bass guitar music, pushing it closer to the style of Pastorius.

For classification, two different approaches have been taken into account, both analyzing a corpus of solo pieces by Pastorius and a corpus of pieces by the bassists Victor Wooten. In the first approach, the pieces have been split into windows of a fixed musical length and 416 different musical features have been extracted from each window. Different window lengths have been tried out (2, 4, 6, 8, 10 and 12 quarter lengths), all with a hop size of 50%. So this constitutes a supervised binary classification task, mapping 416 features to the bassist, Pastorius or Wooten. The idea of classifying a window in based on the assumption, that the style is remarkable even on a local level. Some quarter lengths of music should be sufficient to recognize the artist, not only the whole piece. If this should be used for classifying a whole piece, a simple majority voting across all windows can be done. Gradient Tree Boosting turned out to be a well performing machine learning model for this task. Beside its good performance Decision Tree based learning has another advantage: Feature importance can be estimated and one can attempt to give a kind of comprehensible reason why a specific decision has been taken.

The other approach is training several Markov chains. This is done separately for Pastorius and Wooten, separately for note durations and pitches and for every type of chord that is underlying the span of music used for training the chain. Pitches have been transposed according to the root of the underlying chord for the pitch related Markov chains. There is the possibility of using fixed order Markov chains or variable order Markov chains. After training, the Markov chains can be used for classification straightforwardly: One just has to compute the product of all transition-probabilities of the piece to be considered – respectively for the Pastorius and the Wooten Chains. The higher product indicates the estimated artist. See figure 1 for a graphical outline of the classification.

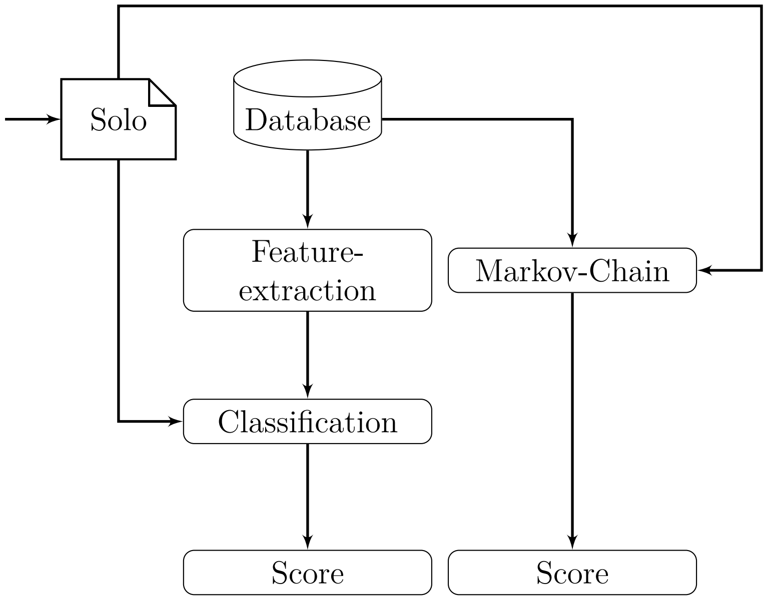


Figure 1: Model for classification

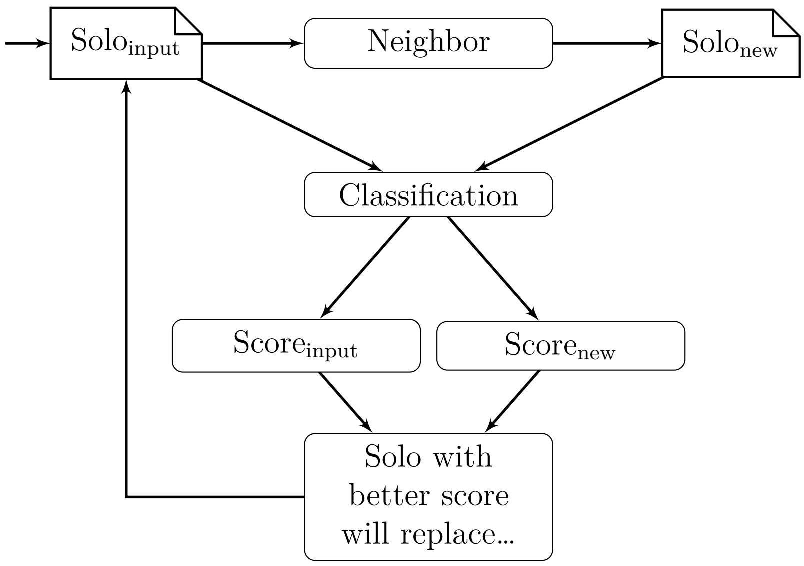


Figure 2: Model for Jaconizer

This classification result can be used as a score within a local search procedure to get a piece of music that has a higher score, i.e. that appears more like Jaco Pastorius according to the classification. This modification procedure is called "Jaconizer" here. See figure 2 for a graphical outline. Local search is an iterative procedure. In each step a small random modification is made to the music according to pitch or rhythm. When classification reports it to be closer to Pastorius than the step before, this version replaces the one before. Otherwise this step is rejected. This is repeated until a time threshold is surpassed or until there are no possible changes that improve the score anymore.

Although figure 1 suggests that there are two scores to be taken into account, actually there are four ones, some of them already compound of different scores. The scores are described in more detail in section 3.3 on pp. 49 et seqq., but the point to notice here is, that this local search is a multi-objective optimization, so it is not straightforward to decide which of two versions is more optimal, because some of the scores could be higher and some could be lower. So, for the local search, only Pareto improvements are accepted: steps that improve at least one score without making any other one worse.

2 Classification

2.1 Data corpus

In this chapter a novel approach of classifying the music of Pastorius is presented: Firstly the investigated data corpus is described, then customly developed feature extractors, useful for classification, are explained. Afterwards two different approaches for classification are described in detail – Gradient Tree Boosting as well as Markov models – and finally classification results are presented.

For analyzing the musical style a corpus of pieces of music is needed, consisting of music by Pastorius and at least one counterexample. For getting started, initially user transcriptions in form of Guitar Pro Files, a popular file format among guitar and bass guitar musicians, related to Jaco Pastorius or Marcus Miller have been searched for in the world wide web, regardless of the music being solo or accompaniment and monophonic or polyphonic. Those files are easily and quickly available and made an early starting with the coding possible. See appendix E.3 on p. 80 for a list of used pieces. The indiscrimination of choosing the pieces as well as the fluctuating quality of the user transcriptions led to unsatisfactory classification results.

Then better transcriptions have been involved: The transcriptions of solo pieces by Pastorius-expert Sean Malone[24] as well as transcriptions of pieces of Victor Wooten by himself[25] have been transcribed into a machine readable format. For Wooten the corpus isn't restricted to solo pieces, also accompaniment is added. It nevertheless can operate as a counterexample. Solo pieces would be beneficial, but it is not so easy to find a monophonic corpus of electric bass guitar solos, transcribed in high quality. Wooten has been chosen, because his style, although influenced by Pastorius, is quite different: compared to him his playing can be described as more diatonic, rather steady in rhythm and more closer to Funk.

They first have been transcribed into Lilypond[26] with chord annotations as simple markup. This can be transformed into Lilypond's internal Scheme representation.[27] A custom Lisp parser has been written for converting the needed subset of Lilypond's Scheme code into Python code, that could be read by the music21-library.[28] This

---

[24] Sean Malone: *A Portrait of Jaco. The Solo Collection*, Milwaukee 2002 (henceforth cited as *Jaco Portrait*).

[25] Victor Wooten: *The Best of Victor Wooten. Transcribed by Victor Wooten*, Milwaukee 2003.

[26] `http://www.lilypond.org`

[27] This can be done easily with Lilypond's `\displayMusic` function.

[28] `http://web.mit.edu/music21`

one finally can output MusicXML,[29] the most popular exchange format for symbolic music. Although that seems to be a bit roundabout, it was a solution for transcribing the music as well as the chord annotation in a single workflow. See appendix E.4 on p. 81 for a list of pieces in the final corpus. It also includes some pieces by Charlie Parker in versions arranged for bass guitar.[30] These pieces are not used for training, but as a test case for some further investigations concerning the influence of Parker on Pastorius.

All music examples in this thesis are newly typesetted versions of the publications mentioned here (Malone for Pastorius, Wooten for himself, Shellard for Parker), when not mentioned otherwise. The various examples won't be quoted every time.

Depending on the window size different amounts of training examples are in the corpus:

window size (in ql[31])	**Jaco Pastorius**	**Victor Wooten**	**Charlie Parker**
2	1977	3465	1141
4	1050	1765	617
6	718	1181	420
8	542	886	315
10	433	709	251
12	358	592	211

Figure 3: Amount of windows per bassist for different window sizes.

2.2 Feature extraction

2.2.1 music21's feature extractors

music21 provides some rich feature extraction facilities,[32] implementing both native feature extractors and the ones of jSymbolic, a standard-set of features for sym-

29 `http://www.musicxml.com`

30 Martin Shellard: *Charlie Parker for Bass. 20 Heads & Sax Solos Arranged for Electric Bass with Tab*, Milwaukee 2014. The book doesn't mention the author being Martin Shellard, but see appendix D on p. 77 for a clarification of the authorship.

31 Short for quarter length. A quarter note has 1 ql, a half note has 2 ql, an eighth-triplet has 1/3 ql, etc.

32 Michael S. Cuthbert/Christopher Ariza/Lisa Friedland: *Feature Extraction and Machine Learning on Symbolic Music using the music21 Toolkit*, in: Anssi Klapuri/Colby Lei-

bolic music classification.[33] Although some of jSymbolic's extractors have not been implemented yet and some are not applicable for the use case discussed here (e.g. `ViolinFractionFeature`, the "Fraction of Note Ons belonging to violin patches"[34]), there is a considerable amount of feature extractors by music21 used in this project: 6 native ones (each outputting a single dimension) and 47 ones from jSymbolic, outputting 324 dimensions. See appendix F on p. 82 for a list of used features.

2.2.2 Custom Feature extractors

2.2.2.1 Preface

In this section all custom feature extractors are explained and examples are given for the sake of clarification. Some notes regarding notation: The single symbol f refers to the feature currently explained. A succession of notes is called n and its single notes are n_i with $1 \leq i \leq N$. $dur(n_i)$ means the duration of a note in quarter-length, $pitch(n_i)$ means the pitch of note as midi number.[35]

A feature that is used, but which doesn't need its own subsection is the window position. The relative position of each window within the work of music also forms a piece of information that is only revealing when combined with the other features to capture the evolution of the features across the work of music.

2.2.2.2 SeqsPerNote

Some of the custom features use the concept of sequences of notes with common direction. Common direction means either successively ascending or descending in pitch. Unisons are considered "neutral" and always count into the current direction.

This feature presents the ratio of the number of sequences and the amount of notes within the piece of music to be considered.

Figure 4 shows the sequences with brackets above the corresponding notes: lower brackets for descending sequences and upper brackets for ascending sequences. In this example there are two different bars with a different amount of sequences and notes. The first bar contains 15 notes and presents 7 sequences, so $f = 7/15 \approx 0.467$

der (eds.): *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011, pp. 387–392, url: `http://ismir2011.ismir.net/papers/PS3-6.pdf`.

33 Cory McKay: *Automatic Music Classification with jMIR*, online (accessed 4-August-2015), PhD thesis, Montreal, 2010, url: `http://jmir.sourceforge.net/publications/PhD_Dissertation_2010.pdf` (henceforth cited as *jMIR*), especially section 4.5, pp. 204 et seqq.

34 Ibid., p. 208.

35 With 60 being middle C.

whereas the second bar contains 13 notes and presents only 3 sequences, so $f = {}^{3}/_{13} \approx 0.231$.

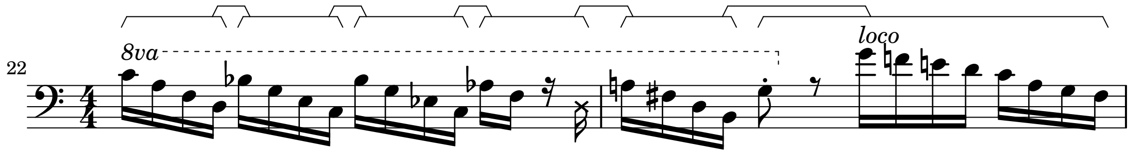


Figure 4: Bar 22–23 of Pastorius' *Port of Entry*.

2.2.2.3 DominantChordType

This feature doesn't describe the style of the bass player, but can be valuable when it is related to other features. The kinds of chords[36] found in the corpus are enumerated and the feature reflects the number related to the most dominant chord $\hat{C}$ to be found. It should be noted that this is a nominal scale.

The following table shows the numbers for the pitch classes of each chord in the corpus.

f	$\hat{C}_{pitchclasses}$	f	$\hat{C}_{pitchclasses}$	f	$\hat{C}_{pitchclasses}$
0	$\varnothing$	1	$\{0\ 1\ 3\ 4\ 7\ \mathrm{t}\}$	2	$\{0\ 1\ 3\ 7\ \mathrm{t}\}$
3	$\{0\ 1\ 4\ 6\ \mathrm{t}\}$	4	$\{0\ 1\ 4\ 7\ \mathrm{t}\}$	5	$\{0\ 2\ 3\ 4\ 7\ \mathrm{t}\}$
6	$\{0\ 2\ 4\ 7\}$	7	$\{0\ 2\ 4\ 7\ \mathrm{t}\}$	8	$\{0\ 2\ 7\}$
9	$\{0\ 3\ 4\ 7\ \mathrm{t}\}$	t	$\{0\ 3\ 4\ 7\ \mathrm{e}\}$	e	$\{0\ 3\ 4\ 8\ \mathrm{t}\}$
12	$\{0\ 3\ 5\ \mathrm{t}\}$	13	$\{0\ 3\ 5\ 7\ \mathrm{t}\}$	14	$\{0\ 3\ 6\}$
15	$\{0\ 3\ 6\ \mathrm{t}\}$	16	$\{0\ 3\ 7\}$	17	$\{0\ 3\ 7\ \mathrm{t}\}$
18	$\{0\ 4\ 5\ 7\}$	19	$\{0\ 4\ 6\ \mathrm{e}\}$	20	$\{0\ 4\ 7\}$
21	$\{0\ 4\ 7\ \mathrm{t}\}$	22	$\{0\ 4\ 7\ \mathrm{e}\}$	23	$\{0\ 7\}$
24	$\{0\ 7\ \mathrm{t}\}$				

Figure 5: Value of DominantChordType for the pitch classes of each chord in the corpus.

Some examples: $f = 16$ means there is a plain minor chord, $f = 21$ there is a dominant seventh chord and $f = 22$ means there is a major seventh chord.

[36] Here the kind of a chord is defined by its pitch classes, when the chord is transposed to C.

2.2.2.4 DurationStd

Here the standard deviation of durations is calculated hence the diversity in used duration is measured. Recall that the standard deviation is the square root of average squared deviations from the mean:

$$f = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(dur(n_i) - \mu)^2}\ , \text{ where } \mu = \frac{1}{N}\sum_{i=1}^{N} dur(n_i) \tag{1}$$

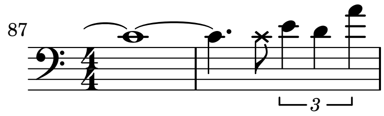


Figure 6: Bar 87–88 of Pastorius' *(Used to be a) Cha Cha.*

In the first bar there is only a single note, so the standard deviation is 0. In the next bar there is the following succession of durations: $\left({}^3\!/\!_2 \quad {}^1\!/\!_2 \quad {}^2\!/\!_3 \quad {}^2\!/\!_3 \quad {}^2\!/\!_3\right)$, so $f = \sqrt{{}^{19}\!/\!_{150}} \approx 0.356$.

2.2.2.5 HarmonicAnticipationDelay

In Jazz, harmonic anticipation is quite common:[37] Notes of the chord that follows, are used before that chord is reached. In addition to that, I call the practice of using notes from the previous chord within the current chord "harmonic delay".

The duration of notes, that match the last resp. following chord without interruption, are counted – backwards in the case of anticipation and in normal order in the case of delay. The duration of those notes is related to the sum of the durations of all notes – without rests.

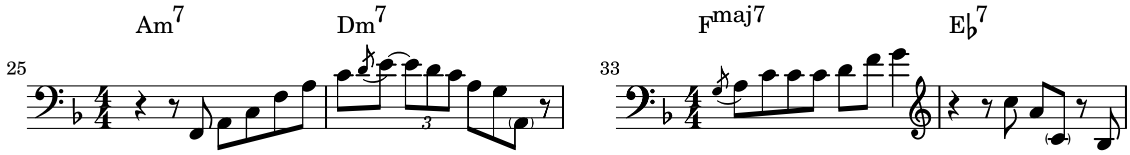


Figure 7: Bar 25–26 and 33–34 of Pastorius' *Days of Wine and Roses*

On the left side there is an example of a full bar being anticipated, hence $f = 1$: Bar 25 shows a broken triad of Dm^7 without its root. a and c is shared by both

[37] Richard Lawn/Jeffrey Hellmer: *Jazz Theory and Practice*, Los Angeles 1996 (henceforth cited as *Jazz Theory and Practice*), p. 77.

chords, but f indicates the full bar anticipates the next chord. In bar 33 the last note g can be seen as anticipated (although it matches both chords), but the second last note f does not match $E\flat^7$, so no further notes are taken into account. For this bar $f = 1/4$.

In contrast, only the first note of bar 26 can be seen as a delay, so $f = 1/7 \approx 0.143$. Bar 34 shows a more suggestive example: All notes except the last one match F^{maj7}, so $f = 3/4$.

2.2.2.6 IntervalDurationAutoCorrelationInfos

First the autocorrelation function of the intervals between the notes (without rests) as well as the duration of the notes (with rests) are computed. Then the average, maximum,[38] minimum and standard deviation of both autocorrelation functions is returned. So this feature represents an 8-dimensional vector that is correlated with motivic similarity. For getting an undisturbed correlation measure, a couple of additional calculations have to be taken into account:

- For dissimilar signals negative and positive values should cancel out. So before doing the autocorrelation the values have to be centered to zero, i.e. the mean has to be subtracted.[39]
- The autocorrelation has to be divided by the number of notes to be invariant to different amounts of notes to be compared.
- For not favoring the beginning of the piece of music to be considered, circular correlation is used.[40]

---

[38] For computing the maximum, the first value of the autocorrelation is not taken into account. The first value corresponds to a comparison of the values with themselves hence there is always maximum similarity.

[39] Although it is not a standard procedure, Orfanidis also points out, for computing the autocorrelation function of a random signal the mean should be zero: Sophocles J. Orfanidis: *Introduction to Signal Processing*, online (accessed 2-July-2015), 2010, URL: `http://www.ece.rutgers.edu/~orfanidi/intro2sp/`, p. 713.

[40] The relation between correlation and circular correlation is equivalent to the relation of convolution and circular convolution. The latter one is more frequently described in literature, e.g. ibid., p. 515 et seq.
For circular correlation holds:

$$\mathcal{F}^{-1}(X^*Y) = \sum_{l=1}^{N} x_l^* y_{(n+l) \bmod N}$$

$$R_{nn}(l) = \frac{1}{N}\sum_{i=1}^{N} \bar{n}_i \bar{n}_{(i+l) \bmod N}, \text{ where } \bar{n} = n - \frac{1}{N}\sum_{i=1}^{N} n_i \tag{2}$$

Note that the equation could be formulated twice, for $dur(n_i)$ as well as for $pitch(n_{i+1}) - pitch(n_i)$, in the latter case up to $N-1$.

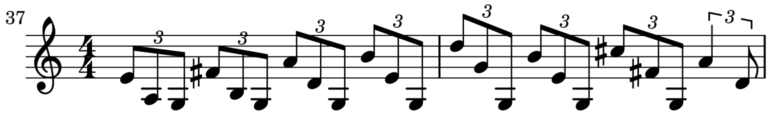


Figure 8: Bar 37–38 of Pastorius' *Havona*

This example is of high motivic similarity. Each triplet presents a perfect fifth followed by G_3 which acts like a pedal point.[41] There is no literal motivic repetition, but the motif is varied constantly and the autocorrelation is able to capture this property.

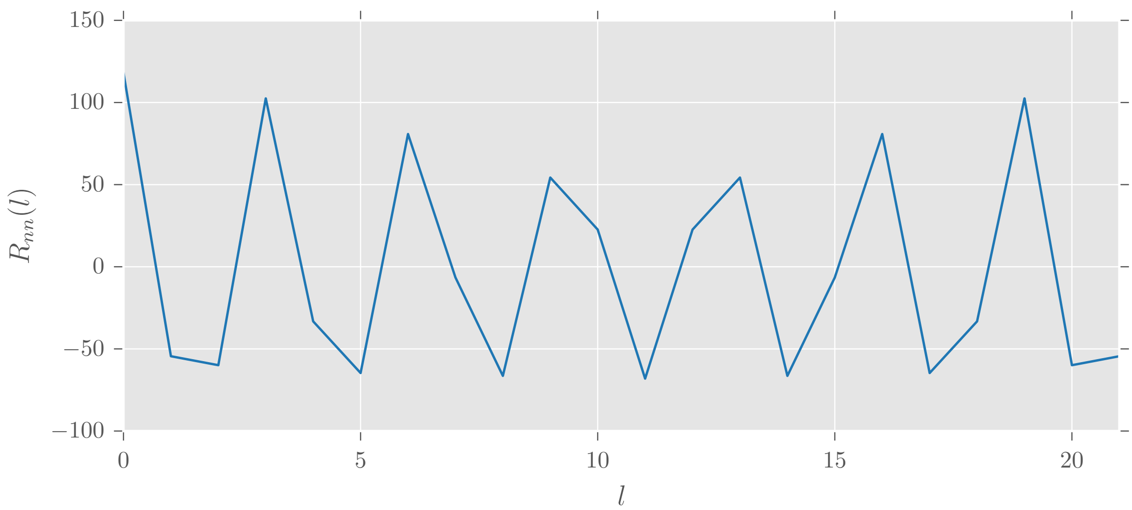


Figure 9: Autocorrelation function of the intervals of figure 8.

In figure 9 it is clearly visible by the high values of the autocorrelation function that every third note there is some motivic repetition.

[41] The intervals of this example can be represented as

$(\; -7 \; -2 \; 11 \; -7 \; -4 \; 14 \; -7 \; -7 \; 16 \; -7 \; -9 \; 19 \; -7 \; -12 \; 16 \; -7 \; -9 \; 18 \; -7 \; -11 \; 14 \; -7 \;)$

and after subtracting the mean approximately

$(\; -6.9 \; -1.9 \; 11.1 \; -6.9 \; -3.9 \; 14.1 \; -6.9 \; -6.9 \; 16.1 \; -6.9 \; -8.9 \; 19.1 \; -6.9 \; -11.9 \; 16.1 \; -6.9 \; -8.9 \; 18.1 \; -6.9 \; -10.9 \; 14.1 \; -6.9 \;)$.

2.2.2.7 MostCommonSeqLen

This feature presents the most common length of the sequences of notes with common direction.[42] The group length of 2 is excluded when finding the most frequent one.

See figure 10 for a clear example of Pastorius with the most common sequence length of 4. All but the last groups have the length of 4.

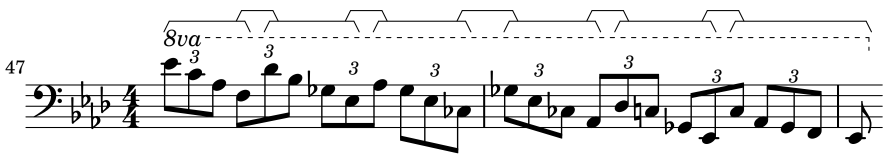


Figure 10: Bar 47–48 of Pastorius' *Donna Lee*.

For another obvious example see figure 8, where $f = 3$.

2.2.2.8 MostCommonSeqDirection

This feature returns the direction of the most common sequence length:[43] -1 for descending, 1 for ascending, or 0 if there isn't any interval within the piece of music to be considered. So for both figure 10 and figure 8 $f = -1$.

2.2.2.9 MostCommonSeqIntervalInfos

For each sequence having the most frequent group length, the interval between the first and the last note is considered. From this succession of intervals, the average, maximum, minimum and standard deviation is returned. So this feature is correlated with the steadiness or diversity of the sequences.

Figure 11 shows the frame intervals of the sequences of length 4 for figure 10. Its intervals are $\begin{pmatrix}-10 & -10 & -9 & -10 & -10\end{pmatrix}$ and so $f = \begin{pmatrix}-{}^{49}\!/_{5} & -9 & -10 & \sqrt{{}^{4}\!/_{25}}\end{pmatrix}$

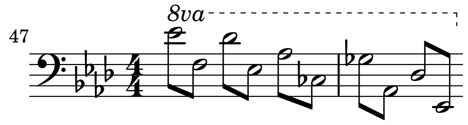


Figure 11: Bar 47–48 of Pastorius' *Donna Lee*, frame intervals of the sequences of length 4.

[42] See section *SeqsPerNote* on p. 11 for more elaboration on sequences in this context.
[43] See directly above for a description of the most common sequence length.

2.2.2.10 NonchordNoteProportion

This feature relates the duration of notes, that don't fit into the underlying harmony to the duration of all notes. So passing notes, neighbor notes, chord substitutions and the like contribute to high values of the feature.

Figure 12: Bar 1–3 of Pastorius' *Days of Wine and Roses*.

In this example Pastorius greatly varies this feature. Especially in the second bar $E\flat 7$ is rather substituted by $E7$ – no note matches the harmony, so the feature would be 1.0 for this bar. The first bar is rather inside the harmony, but there is the emphasized passing note d and the anticipation $e\flat$, so $1^1/_2$ ql of $2^2/_3$ ql doesn't match the harmony, so $f = {}^9/_{16}$.

2.2.2.11 NoteRestRatios

This feature presents the ratio of the duration of all notes and all rests. Consider the 3 bars of figure 12: The notes take $6^2/_3$ ql and the rests take $5^1/_3$ ql, so $f = {}^5/_4$.

2.2.2.12 PitchClassHistogramRelativeToChord

A pitch class histogram relative to the root of the current harmony is a 12-dimensional vector and captures important information about how the musician plays within the chord.

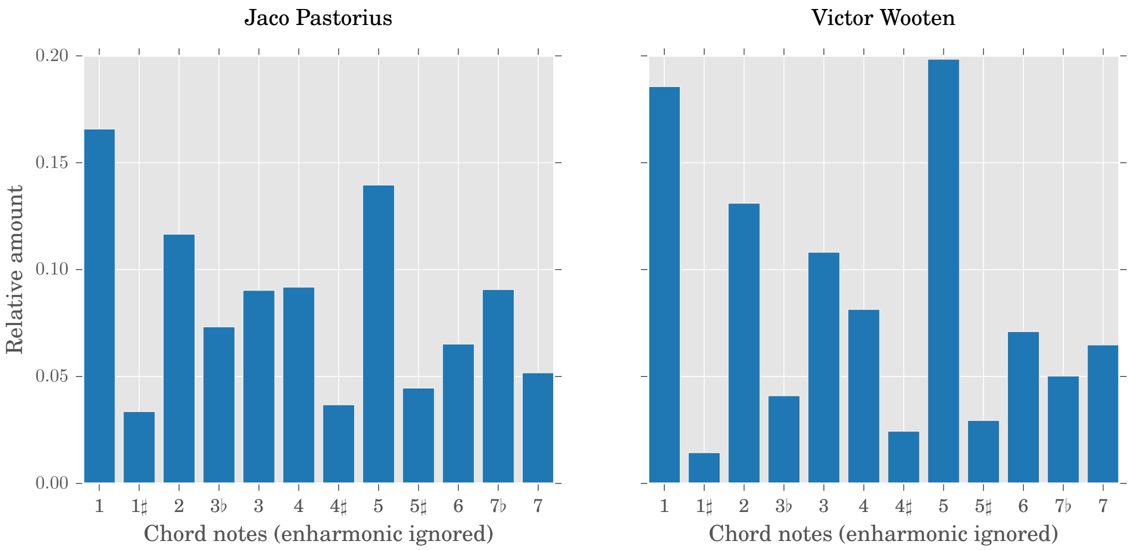


Figure 13: Bar chart of PitchClassHistogramRelativeToChord for all transcribed titles of Pastorius and Wooten.

The plots in figure 13 already give a shallow impression of the more chromatic style of Pastorius playing. He uses dissonant notes more frequently than Wooten, like the minor ninth or the minor seventh whereas Wooten seems to really like to play the fifth of the chord.

2.2.2.13 PitchClassHistogramRelativeToChordForBeats

In this case not only one, but four pitch class histograms relative to the root of the current harmony are calculated – for notes on the 1st, 2nd, 3rd or 4th beat. So it is a 48-dimensional vector.

See figure 14 for some plots. Pastorius seems to use the pitches rather uniformly on all beats – although there are slight preferences, e.g. playing the root on beat 1 or 3, playing the tritone more likely on later beats and playing the minor seventh on the first beat. Preferences for pitches on particular beats are way more dominant for Wooten, e.g. playing the root or the major third on the first beat, the fifth on the second beat or playing the major seventh on the 4th beat.

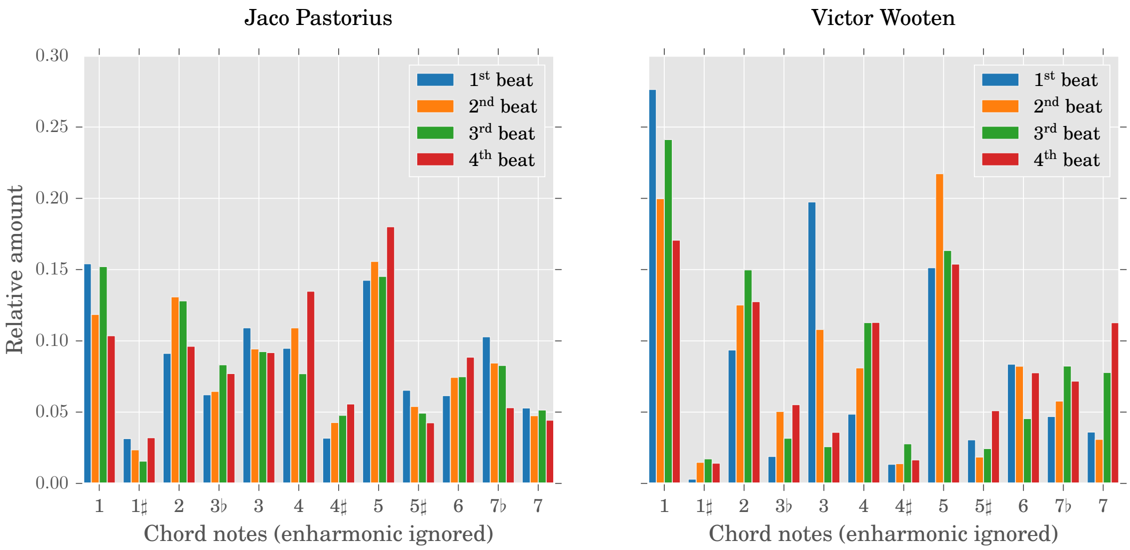


Figure 14: Bar chart of PitchClassHistogramRelativeToChordForBeats for all transcribed titles of Pastorius and Wooten.

2.2.2.14 PitchStd

This feature presents the standard deviation of the used pitches so it is a measure of how the pitch is varied and it complements music21's built-in PitchVariety.[44]

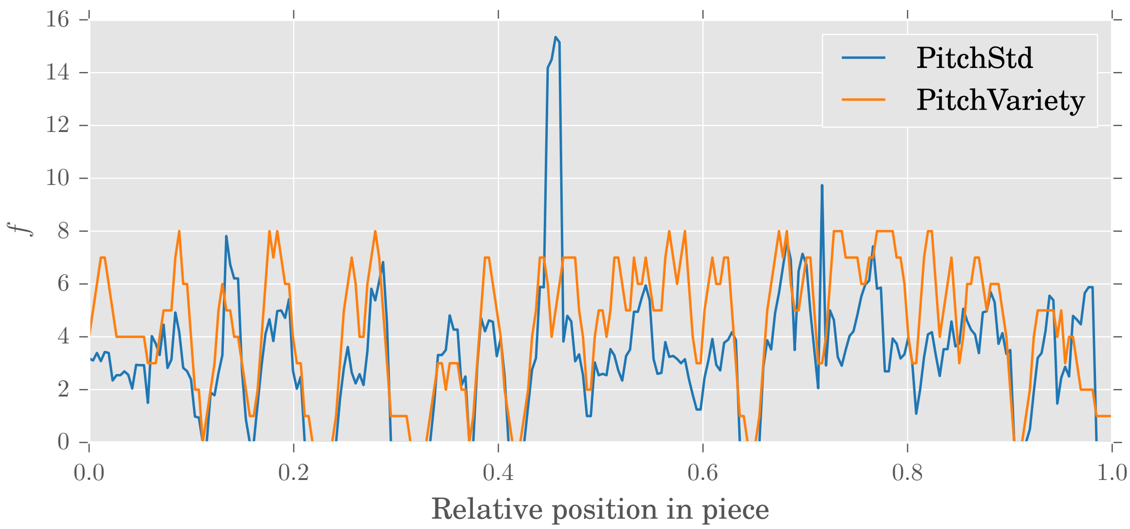


Figure 15: Development of PitchStd and PitchVariety over the full piece of Pastorius' *Punk Jazz* (window size of 4 ql a hop size of 1 ql)

[44] That one only counts how many pitches are used once, so it doesn't take into account how different the pitches are.

In figure 15 it is visible that both values are somehow related, but there is no real correlation. If the PitchVariety is 1 than the PitchStd must be 0. But let's consider the interesting relative position at about 0.45 – there both features seem to have contrary tendencies.

Figure 16: Bar 30 of Pastorius' *Punk Jazz*.

The reason for this can easily be seen by looking at the music. A very high pitched harmonic increases the standard deviation because it is very different from the average pitch in this bar. That is something that cannot be captured by PitchVariety.[45]

2.2.2.15 QuarterLenSeqLenRatio

This feature relates the value of the most common quarter length[46] and the MostCommonSeqLen. So it tries to model a musical characteristic that is one of Pastorius' most striking ones according to Malone:

> Measure 47 [of *Donna Lee*, author's note] contains the first occurrence of what would become a Pastorius trademark: eighth-note triplets in four-note groups, outlining descending seventh-chord arpeggios. The effect is polyrhythmic – the feeling of two separate pulses within the bar that don't share an equal division. [...] As we will see, Jaco utilizes this same technique (including groupings of five) in many of his solos.[47]

This feature is calculated by taking the fractional part of the quotient of the MostCommonSeqLen and the denominator of most common quarter length.

$$f = \frac{f_{MostCommonSeqLen}}{b} \bmod 1, \text{ where } \frac{a}{b} = f_{MostCommonNoteQuarterLength}, \quad \gcd(a,b) = 1 \tag{3}$$[48]

[45] The pitches of this bar are $\begin{pmatrix}58 & 50 & 54 & 93 & 58 & 54\end{pmatrix}$, so $f = \sqrt{7565/36} \approx 14.496$.

[46] That is one of music21's built-in native features.

[47] *Jaco Portrait*, p. 6.

[48] gcd means greatest common divisor. This line is just for the purpose to indicate hat a/b is irreducible in lowest terms.

See figure 10 for an example. The most common quarter length of this example is $1/3$ and the MostCommonSeqLen is 4, so $f = 1/3$. In figure 8 Pastorius doesn't show this characteristic – the most common quarter length is $1/3$ as well, but the MostCommonSeqLen is 3, so $f = 0$.

2.2.2.16 WholeToneScaleAmount

This feature returns the ratio of the amount of notes that fits into a whole tone scale. The two possible pitch class sets for a whole tone scale are $\{0 \quad 2 \quad 4 \quad 6 \quad 8 \quad \mathrm{t}\}$ and $\{1 \quad 3 \quad 5 \quad 7 \quad 9 \quad \mathrm{e}\}$ and the bigger ratio of the both is returned.

Figure 17 shows an example of a nearly full whole tone scale usage, so $f = 1.0$ if you don't consider the harmonic chord and if you consider it $f = 6/7$. In the implementation of the described project the chord wouldn't be taken into account.

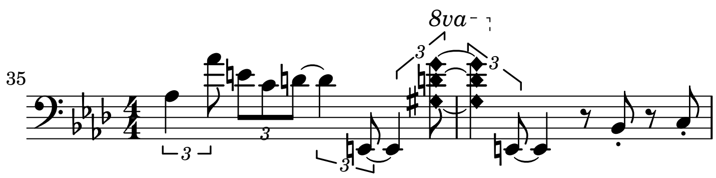


Figure 17: Bars 35–36 of Pastorius' *Donna Lee*.

2.3 Classification of features

2.3.1 General

A model should be trained to get musical features as input variables and predict if the corresponding bassist is Pastorius or not. Different machine learning algorithms have been tried out for finding a good classification. Gradient Tree Boosting proved to be one of the best ones in the scenario of the project described here.[49] In the following, an overview about Gradient Tree Boosting is given, including an overview about Decision Trees. Gradient Tree Boasting utilizes many Decision Trees, so both methods are important in this scope.

[49] For the performance of other classification algorithms with their default-parameters from the scikit-learn library (`http://scikit-learn.org`), see the following table. The conditions for the reports are equivalent with the ones described in section 2.5 (with window size of 6 ql and 3 folds).

In general, we want to find a function f, that receives a feature vector x_i from a feature matrix x and predicts an output variable y_i. For that purpose, we utilize N training examples for fitting the function. x are called the descriptor variables and y is called the dependent variable.

Let's consider a concrete example. Figure 18 shows a subset of the training corpus with window size 6 ql. We have $N = 10$ training examples, each of them a 2-dimensional[50] feature vector and the bassist as the dependent variable.

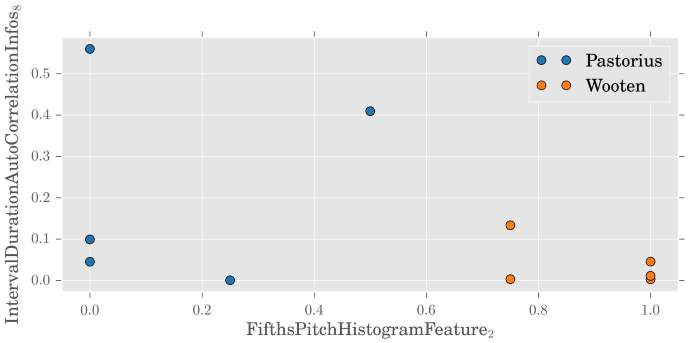
This test data set serves for illustration purposes in the following.[51] Since there are only two descriptor variables, it can easily be plotted, see figure 19.

Support vector machine with stochastic gradient descent training	71.119% (±4.483%)
K-nearest neighbors	80.701% (±0.855%)
Feed-Forward Neural Network*	83.168% (±1.082%)
Decision tree classifier	83.175% (±0.328%)
C-Support Vector Classification	86.941% (±0.681%)
AdaBoost classifier	87.757% (±1.093%)
Extra-trees classifier	87.915% (±0.165%)
Random forest classifier	88.863% (±1.036%)
Gradient Boosting classifier	91.917% (±0.417%)

* This one not from scikit-learn, but from pybrain (`http://www.pybrain.org`). Non-default parameters – topology: input layer: 250, 1st hidden layer: 125, 2nd hidden layer: 83, output layer: 2), learning rate=0.001, weight decay=0.01.

[50] For a description of the features see paragraph 2.2.2.10 on p. 17 and paragraph 2.2.2.9 on p. 16. Subscript 2 means the second value (the maximum) of this 4-dimensional vector.

[51] Note that for this purpose it is balanced between too easy and too complicated, which can be done manually in such a small example set. For the same windows one easily can find another subset of descriptor variables that would be correctly classified even with an unsupervised clustering algorithm, e.g.



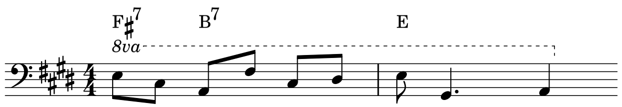
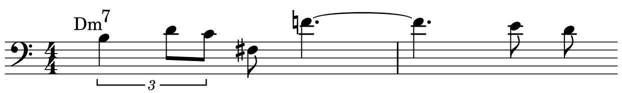
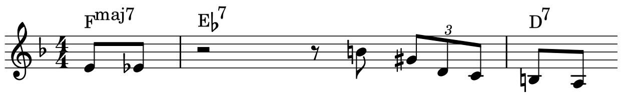
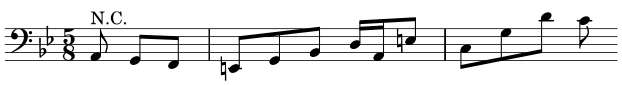
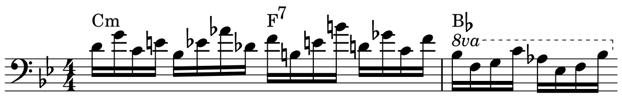
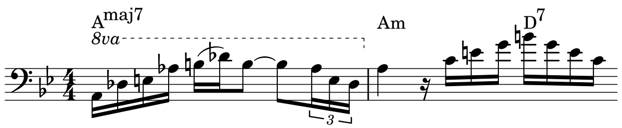
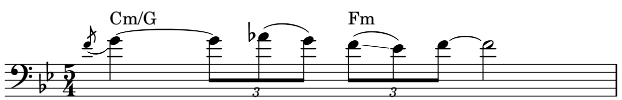
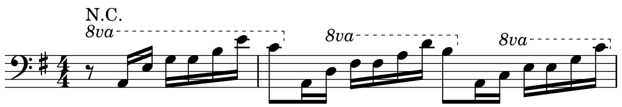
	NonchordNote Proportion	MostCommon SeqInterval Infos_2	Bassist
	0.25	3.	Pastorius [from *Donna Lee*]
	0.28571429	11.	Pastorius [from *(Used To Be A) Cha Cha*]
	0.	-9.	Pastorius [from *Havona*]
	0.	-8.	Pastorius [from *Havona*]
	0.71428571	-14.	Pastorius [from *The Days of Wine and Roses*]
	0.	10.	Wooten [from *Blu-Bop*]
	0.58333333	12.	Wooten [from *Cherokee (Indian Love Song)*]
	0.39130435	-16.	Wooten [from *Cherokee (Indian Love Song)*]
	0.8	3.	Wooten [from *Blu-Bop*]
	0.	29.	Wooten [from *Classical thumb*]

Figure 18: Test data set for illustration purposes

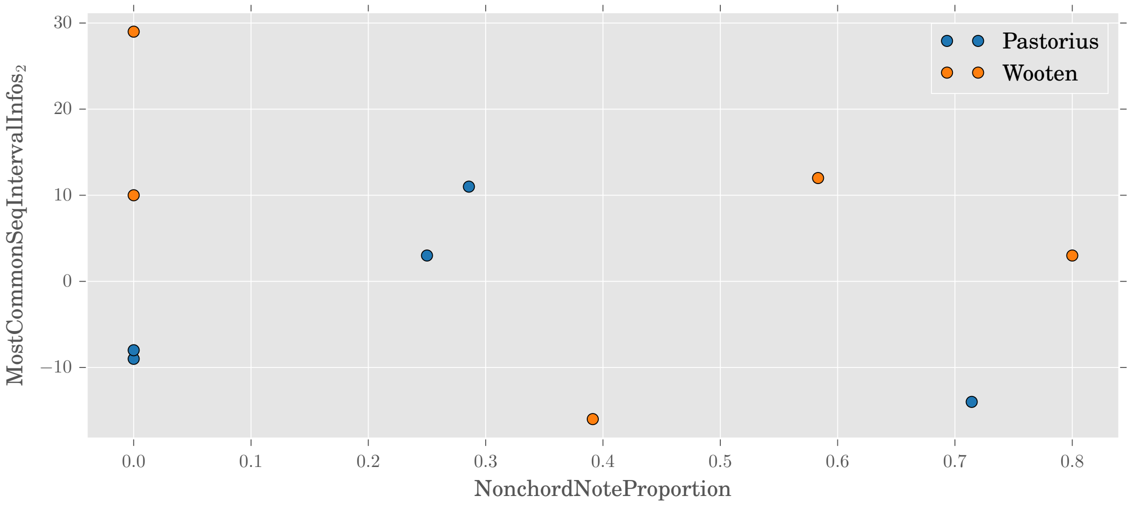


Figure 19: Scatter plot of the test data set.

2.3.2 Decision Trees

Decision Trees segment the domain of the descriptor variables into rectangle-alike regions. A constant (e.g. the index of a class) is assigned to each region, so classifying is just a look-up in the appropriate region. So the question is how to find good segmentations of the domain of the descriptor variables. There exist methods to do this – a commonly used one is CART[52] for regression, which is used in this project.

The main idea is firstly splitting one variable. The constants assigned to the two emerging regions are just the mean of the dependent variable of all training items corresponding to this region. The variable and split-point are chosen to achieve the best fit. Then this process is continued until the training items are perfectly explained or some stopping criteria is achieved. Finally the model has learned a function:

$$f(x) = \sum_{m=1}^{M} c_m \mathbf{1}_{R_m}(x) \tag{4}$$

with

$$c_m = \text{ave}(y_i | x_i \in R_m) \tag{5}$$

[52] Trevor Hasties/Robert Tibshirani/Jerome H. Friedman: *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*, 2nd ed., New York 2009 (henceforth cited as *Statistical Learning*), p. 305.

Here M is the number of regions $R_1, R_2, \ldots, R_M$; c_m is the constant assigned to a particular region, averaging the dependent variable of all training items corresponding to this region; and $\mathbf{1}_{R_m}(x)$ is an indicator function, evaluating to 1 if x is within the region and 0 otherwise.

The open question is, how to find the best splitting variable j and split point s. That can be found by the values that solve

$$(j, s) = \min_{j,s} \big(\min_{c_1} \sum_{x_1 \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_1 \in R_2(j,s)} (y_i - c_2)^2 \big). \tag{6}$$

Both inner minimizations can already be solved, see equation 5. j can be found by brute-force and s can, e.g., be found by sorting all values and trying out the mean values of all neighboring values that have different classes.[53] This process can then be repeated for all regions. The remaining question – how large the tree should be – is related to the value of M. This is a tuning parameter which should be estimated with regards to the data. There are several ways to do that, not discussed here.[54]

If a single decision tree would be used for classification, some adoptions would have to be made[55], but for solving a binary classification task[56] with Gradient Boosting, Regression Trees are used anyway. So decision trees for classification are not used in this project.

Figure 20 shows three decision trees with successive depth learned on the training data. Note that with increasing depth the decision boundary becomes more complex, more fitted to the data and less training items are misclassified. Some remarks to the tree visualizations: Numbers are rounded to 3 decimal places. MCSII and NNP are abbreviations of the feature names.[57] For learning, an integer number was assigned to each bassist as a dependent variable: 0 to Wooten and 1 to Pastorius. So the values of the constant, therefore denote the average of the dependent variable of the corresponding region, can be explained. Mse stands for mean square error, a measure of how many items are misclassified.

$$\text{mse}(x, y) = \frac{1}{N} \sum_{i=1}^{N} (f(x_i) - y_i)^2 \tag{7}$$

[53] Stuart Russell/Peter Norvig: *Artificial Intelligence. A Modern Approach*, 3rd ed., Harlow 2014 (henceforth cited as *Artificial Intelligence*), p. 718.
[54] See *Statistical Learning*, p. 308, and the references mentioned there.
[55] Ibid., pp. 308 et seqq.
[56] That is the case in the project described here: Pastorius or Non-Pastorius.
[57] MCSII stands for MostCommonSeqIntervalInfos and NNP for NonchordNoteProportion, see p. 23.

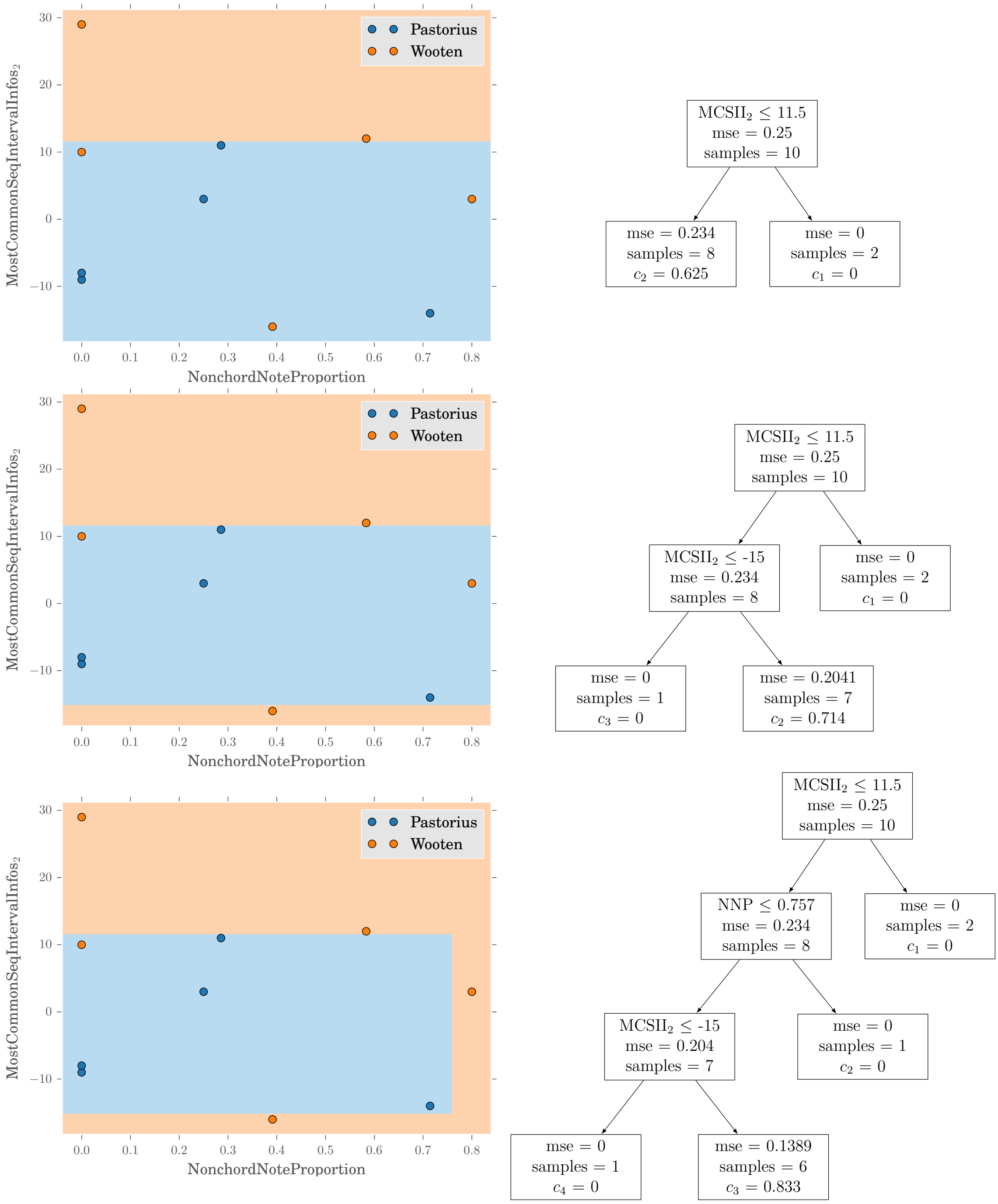


Figure 20: Contour plot (left) and tree visualization (right) of three decision trees, that have been trained on the test data set.[58] The trees have been restricted to different depths: 1 (top), 2 (middle), 3 (below).

2.3.3 Gradient Tree Boosting

The idea of Tree Boosting is instead of using a single tree f, an ensemble of trees F is used, consisting of J simple trees.[59] The idea of an ensemble of small trees is related to the idea of weak learners, classifiers that perform only slightly better than random guessing, but can be transformed into strong learners, classifiers of high accuracy.[60] Instead of learning a highly accurate prediction rule, many rough rules of thumb are learned, combined being as strong or even stronger as the highly accurate one.

If you consider equation 4, a tree is defined by its regions R_m and its constants c_m. They form the parameters of the tree $\Theta = \{R_m, c_m\}_1^M$. When specifying a tree $f(x)$ with its parameters, we call it $f(x;\Theta)$. This way an ensemble of trees can be defined, where the output of each tree is scaled by a factor β:

$$F(x) = \sum_{j=1}^{J} \beta_j f(x;\Theta_j). \tag{8}$$

So the task is finding all parameters $\{\beta_j, \Theta_j\}_1^J$. After learning the parameters for the initial tree, f_1 can be found as usual (see previous section) and the parameters for the next trees can be found by minimizing a loss function L in a stagewise approach. In equation 6 the loss function is the squared difference, but other loss functions are possible.

$$(\beta_j, \Theta_j) = \arg\min_{\Theta,\beta} \sum_{i=1}^{N} L\big(y_i, F_{j-1}(x_i) + \beta f(x_i;\Theta)\big) \tag{9}$$

where

$$F_j(x) = F_{j-1}(x) + \beta_j f(x;\Theta_j). \tag{10}$$

This minimization can be quite difficult. Instead of solving it formally, one tries to come closer to a form similar to equation 6. But not to be restricted to the squared

58 See p. 23 for details.

59 Jerome H. Friedman: *Greedy Function Approximation: A Gradient Boosting Machine*, in: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232 (henceforth cited as *Greedy Approximation*), Jane Elith/John R. Leathwick/Trevor Hastie: *A working guide to boosted regression trees*, in: *Journal of Animal Ecology* 77.4 (2008), pp. 802–813 and *Statistical Learning*, pp. 353 et seqq.

60 For further details on this see Robert E. Schapir: *The Strength of Weak Learnability*, in: *Machine Learning* 5.2 (1990), pp. 197–227.

difference, something more general is needed: The negative gradient of all training items with respect to the loss function can be used as a "pseudo-response" $\tilde{y_i}$ for training the next tree.

$$\tilde{y_i} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right]_{F(x)=F_{j-1}(x)} \tag{11}$$

As a loss function, deviance turned out to give the best results for the project described here:[61]

$$L(y,x) = -\frac{2}{N}\sum_{i=1}^{N} y_i f(x_i) - \log(1 + \mathrm{e}^{f(x_i)}) \tag{12}$$

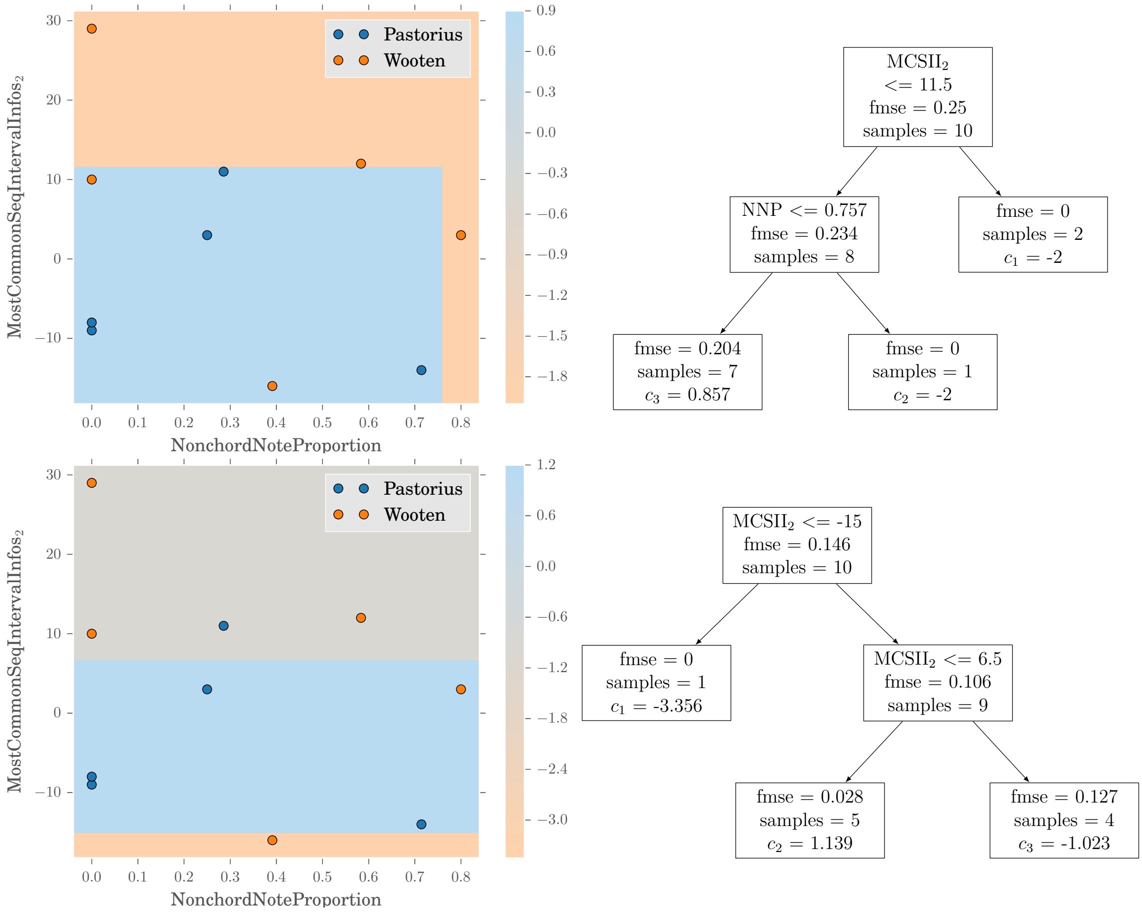
This response is used for learning the parameters of the next tree. Sometimes this is also called Bernoulli deviance.[62] Tweaking some further parameters improved the results.

- The number of boosting iterations J is 800.
- The maximum size for each tree is restricted to 4.
- The learning rate, a constant factor that is multiplied by the output of all trees, is 0.45. (That is relatively high compared with common values.[63])
- There must be at least 3 training examples in each leaf node of a tree.
- For each splitting point only half of the overall number of features are considered at maximum.

Again, let's illustrate the classification process with the test data set from p. 23. A Gradient Boosting Classifier with three Decision Trees has been learned, each of them restricted to the depth of 2. Figure 21 shows those trees. There is a minor difference between the regression trees shown in the previous section: it uses fmse

[61] For more details on loss functions see *Greedy Approximation*, p. 9 and *Statistical Learning*, pp. 346 et seqq. and 360.
[62] This term and some additional information about this loss function, like its gradient, is to be found in Greg Ridgeway: *Generalized Boosted Models. A guide to the gbm package*, 2012, URL: `https://github.com/harrysouthworth/gbm/blob/master/inst/doc/gbm.pdf`, gradually developed online paper for the R gbm package, version from May 23, 2012, p. 10.
[63] *Statistical Learning*, p. 365.

instead of mse as split criterion. Fmse means Friedman mean square error and is a slightly improved version of equation 7.[64] It shouldn't be confused with deviance: Both are used as some kind of loss function, but fmse is used for building a single tree whereas deviance is used to build up the ensemble of trees.



[64] See *Greedy Approximation*, p. 12 (equation 35).

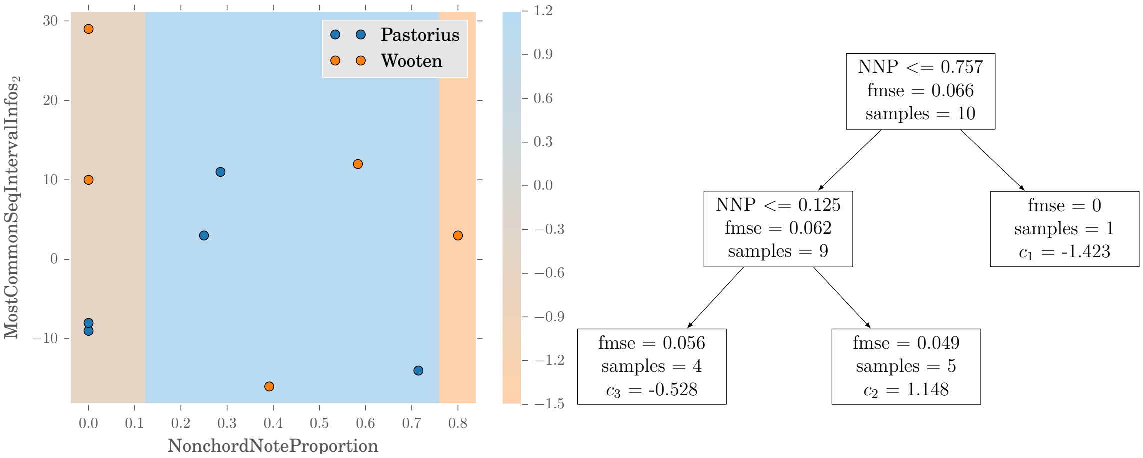


Figure 21: Three Decision Trees of the Gradient Boosting Classifier after training on the test data.[65] Contour plots (left) and tree visualizations (right).

When summing all three trees, we get decision values, that can be converted into probabilities, like this:

$$
\begin{aligned}
P(x = \text{Pastorius}) &= \frac{1}{1 + \mathrm{e}^{\sum_{j=1}^{J} f(x)}} \\
P(x = \text{Wooten}) &= 1 - P(x = \text{Pastorius})
\end{aligned}
\tag{13}
$$

Using these probabilities a binary decision for class estimates can be made.

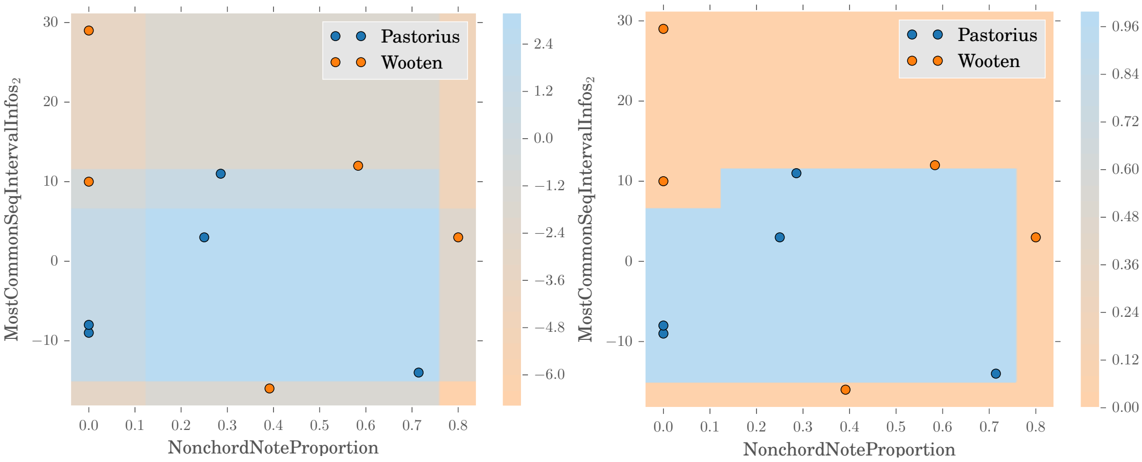


Figure 22: Contour plots of the summed decision trees (left) and class estimates (right) of the Gradient boosting classifier.

[65] See p. 23 for details.

For the sake of illustration all 10 training examples have been used for training. When doing a real training, in practice not all examples are used – some are held out for validating the model on unseen data. For investigating how all training items can be classified when they have not been used for training, this whole process is repeated at least 3 times: Splitting the data set in 3 portions (each reflecting the same proportion of classes as the whole) and then one can do training and validating 3 times: each time leaving out one portion, training on the other portions and validating on the one left out. This process is called stratified k-folds cross validation (k=3 when splitting into 3 portions, but more portions are possible).[66]

2.4 Markov model

A Markov model is a generative model that takes a different approach compared the to previous section. Instead of calculating abstract features, the raw note sequences are involved in learning the model: The probabilities of sequences of a specific length are estimated. In note sequences, each event contains at least two pieces of information: The pitch and the note duration. So two Markov models can be made: One for pitch and one for note durations. There would also be the possibility for accepting pitch and duration to be a single compound event, but this would make all events more rare and that has some drawbacks described below.

Firstly let's describe Markov models more formally.[67] Markov models are useful for processes where discrete events succeed each other and where each event depends on the previous ones. Let's assume there are N distinct states $S_1, S_2, \ldots, S_N$. The process is in a state q_t at time t with $t \in \mathbb{N}_{>0}$. So $q_1 = S_1$ means that at the start of the process it is in the first state. This first state influences the next one yet to come. So the probability of each event occurring depends on its previous states:

$$P(q_t = S_i | q_{t-1} = S_j, \ldots, q_1 = S_k), \text{ with } 1 \leq i, j, k \leq N. \tag{14}$$

The so called first-order Markov assumption means that an event doesn't depend on all but only its immediate previous state:

$$P(q_t = S_i | q_{t-1} = S_j, \ldots, q_1 = S_k) = P(q_t = S_i | q_{t-1} = S_j). \tag{15}$$

66 *Statistical Learning*, pp. 241 et seqq.

67 Ethem Alpaydin: *Introduction to Machine Learning*, 2nd ed., Cambridge and London 2010, p. 363 et seqq.

In general this doesn't hold true, but it is a simplification needed for computational feasibility. Of course one isn't restricted to first-order Markov chains. Also second-, third-, etc. order Markov chains are possible. An Oth-order Markov chains implies that an event only depends on its O immediate predecessors.

For music, even when $\lim_{O \to \infty}$, an Oth-order Markov chain wouldn't hold true, because the probability of a note can even be influenced by its successors. Just think of the climax of a musical phrase that is headed for already some time before. So music isn't a real Markov process but some impressive results nevertheless can be achieved easily, whereby it has a long tradition in music generation.[68]

For the sake of this explanation, let's stay with the first-order model. There is still one further step of simplification: For using it in a generative model, one assumes that the transition probabilities are independent of time:

$$a_{ij} \equiv P(q_t = S_i | q_{t-1} = S_j), \text{ with } a_{ij} \leq 0 \text{ and } \sum_{j=1}^{N} a_{ij} = 1. \tag{16}$$

That way, one can hold all probabilities in a $N \times N$ matrix A, where each row sums to 1. Since each state depends on its predecessor, the model still does not describe its first state. Therefore one needs a further piece of information for completing the model: The initial probabilities Π with $\pi_1, \pi_2, \ldots, \pi_N$ are needed, which state the probabilities for the first state of the process.

$$\pi_i \equiv P(q_1 = S_i), \text{ with } \pi_i \leq 0 \text{ and } \sum_{i=1}^{N} \pi_i = 1. \tag{17}$$

Such a Markov model can be represented graphically, but already small models exhibit quite some complexity in their representations, as seen in figure 23:

[68] See section 1.1 on pp. 1 et seqq. for some references, specifically footnotes 12–14.

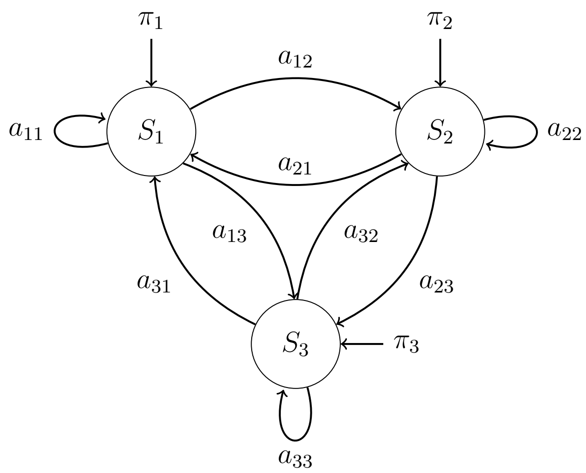


Figure 23: Graphical representation of a first-order Markov chain with 3 different states. Each vertex represents a state and each edge represents a probability – transition probabilities in the case of edges connecting states and initial probabilities in the case of the "virtual edges" coming out of nowhere.

Classification with Markov models is an easy task: At least two models have to be given and than two probabilities can be assigned to an observation sequence $Q = (q_1, q_2, \ldots, q_T)$. The two probabilities state how probable it is that the one or other process has produced such a sequence.

$$P(Q|A,\Pi) = P(q_1) \prod_{t=2}^{T} P(q_t|q_{t-1}) = \pi_{q_1} a_{q_1 q_2} \cdots a_{q_{T-1} q_T} \tag{18}$$

Since we are constantly multiplying numbers smaller or equal to zero, the result can be infinitesimal small in the case of large observation sequences. So it can be beneficial to compute the probabilities in the log space for preventing computational round-off errors. Now they aren't actual probabilities anymore but it is still fine for comparing the results of different models.

$$\log P(Q|A,\Pi) = \log P(q_1) + \sum_{t=2}^{T} \log P(q_t|q_{t-1}) = \log \pi_{q_1} + \log a_{q_1 q_2} + \cdots + \log a_{q_{T-1} q_T} \tag{19}$$

One further consideration has to be taken into account: Longer sequences always tend to have lower probabilities, because there exist more possible long sequences than short ones. So, when comparing sequences of different length, the mean probability can be used.

$$\bar{P}(Q|A,\Pi) = \frac{P(q_1) + \sum_{t=2}^{T} P(q_t|q_{t-1})}{T} = \frac{\pi_{q_1} + a_{q_1 q_2} + \cdots + a_{q_{T-1} q_T}}{T} \tag{20}$$

Training a Markov model from K sequences is straightforward: The probability for a transition from state i to state j is the number of transitions from S_i to S_j, divided by the number of transitions from S_i to any other state.

$$a_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \mathbf{1}_{\{S_i\}}(q_t^k)\, \mathbf{1}_{\{S_j\}}(q_{t+1}^k)}{\sum_{k=1}^{K} \sum_{t=1}^{T^k-1} \mathbf{1}_{\{S_i\}}(q_t^k)} \tag{21}$$

The initial probabilities are even more easy to estimate: The probability for state i being the initial one, is just the number of sequences starting with S_i, divided by the overall numbers of sequences.

$$\pi_i = \frac{\sum_{k=1}^{K} \mathbf{1}_{\{S_i\}}(q_1^k)}{K} \tag{22}$$

When few sequences are involved, one often has to find an alternative way to define the initial probabilities. Two methods are possible:

1. Assign an equal distribution to all initial probabilities: $\pi_i = {}^1\!/\!_N$.
2. Use a zeroth-order Markov chain: $\pi_i = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T^k} \mathbf{1}_{\{S_i\}}(q_t^k)}{\sum_{k=1}^{K} T^k}$

Method 2 is used in the project described here. Let's illustrate it with a short example.

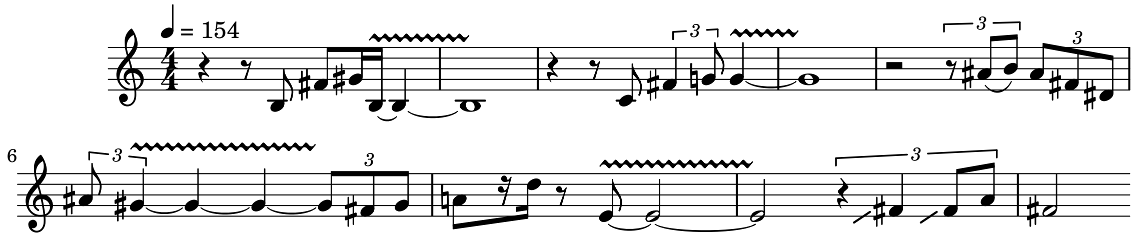


Figure 24: First bars of Pastorius' *Havona*.

The note sequences of fig. 24 can be translated into both pitch sequence Q^p in midi numbers and duration sequence Q^d in ql:

$$Q^p = \begin{array}{ccccccccccccc} (\rho, & 59, & 66, & 68, & 59, & 59, & 59, & \rho, & 60, & 66, & 67, & 67, & 67 \\ \rho, & 70, & 71, & 70, & 66, & 63, & 70, & 68, & 68, & 68, & 68, & 66, & 68 \\ 69, & \rho, & 74, & \rho, & 64, & 64, & 64, & \rho, & 66, & 66, & 69, & 66) & \end{array}$$

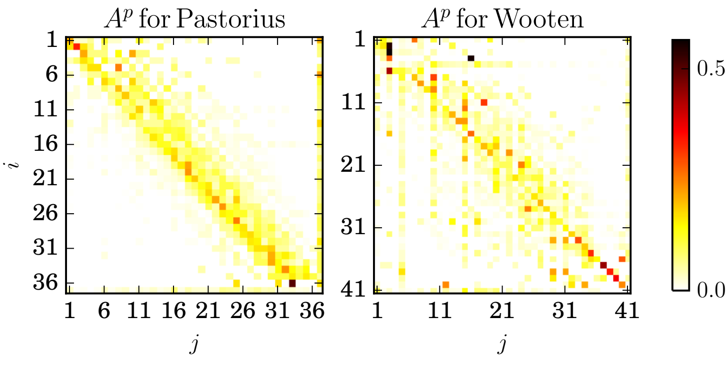
$$Q^d = \begin{array}{ccccccccccccc} (3/2, & 1/2, & 1/2, & 1/4, & 1/4, & 1, & 4, & 3/2, & 1/2, & 2/3, & 1/3, & 1, & 4 \\ 7/3, & 1/3, & 1/3, & 1/3, & 1/3, & 1/3, & 1/3, & 2/3, & 1, & 1, & 1/3, & 1/3, & 1/3 \\ 1/2, & 1/4, & 1/4, & 1/2, & 1/2, & 2, & 2, & 2/3, & 2/3, & 1/3, & 1/3, & 2) & \end{array}$$

Rests are defined having pitch ρ, so they are just considered to be notes with no pitch. Also note, that several musical attributes are not described by those sequences, e.g. tempo, vibrato, bend notes, metre, etc. Ties are also not represented, but successive rests are joined. With those sequences, we easily can set up our set of states S^p for pitch and S^d for duration.

$$\begin{aligned} S^p, &= (59, \; 60, \; 63, \; 64, \; 66, \; 67, \; 68, \; 69, \; 70, \; 71, \; 74, \; \rho) \\ S^d, &= (1/4, \; 1/3, \; 1/2, \; 2/3, \; 1, \; 3/2, \; 2, \; 7/3, \; 4) \end{aligned}$$

By just counting the transitions between states, one can easily estimate the transition probabilities. See figure 25 for a graphical display of the matrices:[69]

[69] The matrices shown here are of course rather sparse because of the small training set. See here the pitch-related first-order Markov transition-probabilities for the whole Pastorius and Wooten corpora. They show much more transitions and offer more alternatives (i.e. there are no probabilities equal to one).



$$A^p = \begin{pmatrix} 1/2 & 0 & 0 & 0 & 1/4 & 0 & 0 & 0 & 0 & 0 & 0 & 1/4 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2/3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 0 & 0 & 1/6 & 0 & 1/6 & 1/6 & 1/3 & 1/6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2/3 & 0 & 0 & 0 & 0 & 0 & 1/3 \\ 1/6 & 0 & 0 & 0 & 1/6 & 0 & 1/2 & 1/6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 & 0 & 1/3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1/6 & 1/6 & 0 & 1/6 & 1/6 & 0 & 0 & 0 & 1/6 & 0 & 1/6 & 0 \end{pmatrix}$$

$$A^d = \begin{pmatrix} 1/2 & 0 & 1/4 & 0 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 2/3 & 1/12 & 1/12 & 1/12 & 0 & 1/12 & 0 & 0 \\ 1/3 & 0 & 1/3 & 1/6 & 0 & 0 & 1/6 & 0 & 0 \\ 0 & 1/2 & 0 & 1/4 & 1/4 & 0 & 0 & 0 & 0 \\ 0 & 1/4 & 0 & 0 & 1/4 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \end{pmatrix}$$

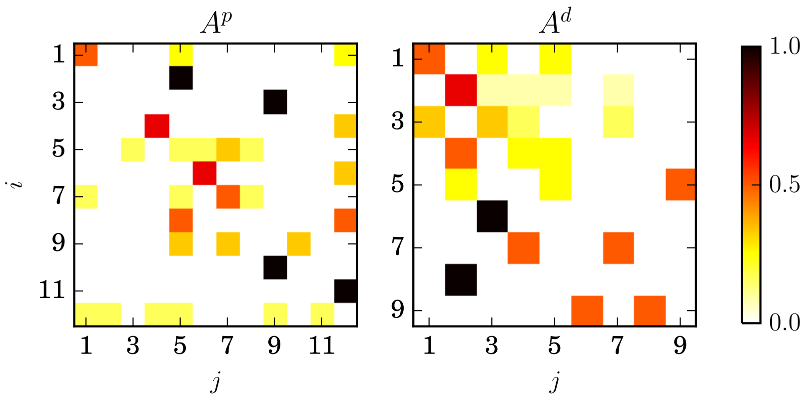


Figure 25: Graphical display of the matrices of transition probabilities.

Such a graphical display can also illustrate the problem of accepting pitch and duration to be a single compound event. With those, one would have a much larger amount of states S^{pd}, each being a tuple of pitch and duration:

$$S^{pd} = ((59,1),\ (59,4),\ (59,{}^{1}/_{2}),(59,{}^{1}/_{4}),(60,{}^{1}/_{2}),(63,{}^{1}/_{3}),(64,2),(64,{}^{1}/_{2}),(66,2),\\ (66,{}^{1}/_{2}),(66,{}^{1}/_{3}),(66,{}^{2}/_{3}),(67,1),\ (67,4),\ (67,{}^{1}/_{3}),(68,1),(68,{}^{1}/_{3}),(68,{}^{1}/_{4}),\\ (68,{}^{2}/_{3}),(69,{}^{1}/_{2}),(69,{}^{1}/_{3}),(70,{}^{1}/_{3}),(71,{}^{1}/_{3}),(74,{}^{1}/_{4}),(\rho,{}^{1}/_{2}),(\rho,{}^{1}/_{4}),\ (\rho,{}^{2}/_{3}),\\ (\rho,{}^{3}/_{2}),\ (\rho,{}^{7}/_{3}))$$

The corresponding matrix of transition probabilities is much more sparse, containing more zeros and ones. Probabilities of one mean that there are no alternatives, so the Markov model is less able to generalize.

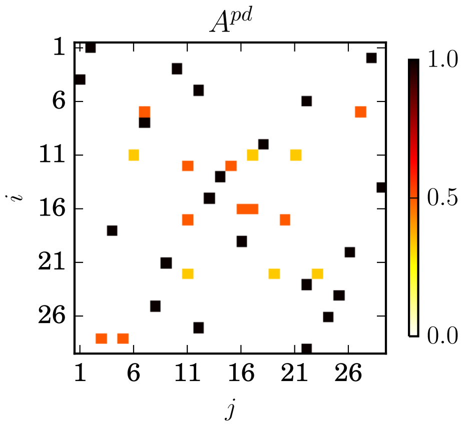


Figure 26: Graphical display of the matrices of transition probabilities for pitch and duration as single compound events.

So let's put the idea of compound events aside and come back to the idea of separate Markov models for pitch and duration. The initial probabilities are computed by the zeroth-order Markov chain:

$$\Pi^{p}, = ({}^{2}/_{19},\ {}^{1}/_{38},\ {}^{1}/_{38},\ {}^{3}/_{38},\ {}^{7}/_{38},\ {}^{3}/_{38},\ {}^{3}/_{19},\ {}^{1}/_{19},\ {}^{3}/_{38},\ {}^{1}/_{38},\ {}^{1}/_{38},\ {}^{3}/_{19})$$
$$\Pi^{d}, = ({}^{2}/_{19},\ {}^{6}/_{19},\ {}^{3}/_{19},\ {}^{2}/_{19},\ {}^{2}/_{19},\ {}^{1}/_{19},\ {}^{3}/_{38},\ {}^{1}/_{38},\ {}^{1}/_{19})$$

With those probabilities one can put a weighted random process into operation to generate two different note sequences, both conforming to the Markov properties described above. Note that such a process can produce illegal sequences, e.g. duration sequences like $({}^{1}/_{2},\ {}^{1}/_{3},\ {}^{1}/_{3},\ {}^{1}/_{2})$, especially for such a low-order Markov model. Nevertheless below are two generated legal sequences:

$$
\begin{aligned}
Q^p_1, &= (68,\ 66,\ 63,\ 70,\ 66,\ 67,\ 67,\ \rho,\ 66,\ 66,\ 63,\ 70,\ 71,\ 70,\ 71,\ 70,\\
&\quad 71,\ 70,\ 68)\\
Q^d_1, &= (2,\ 2,\ \nicefrac{2}{3},\ \nicefrac{1}{3},\ \nicefrac{1}{3},\ \nicefrac{1}{3},\ \nicefrac{1}{3},\ \nicefrac{1}{2},\ \nicefrac{1}{2},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ 1,\ 4,\ \nicefrac{3}{2},\\
&\quad \nicefrac{1}{2},\ \nicefrac{2}{3},\ \nicefrac{1}{3})\\
Q^p_2, &= (68,\ 66,\ 63,\ 70,\ 68,\ 68,\ 68,\ 68,\ 68,\ 68,\ 59,\ 66,\ 69,\ \rho,\ 64,\ \rho,\ 66)\\
Q^d_2, &= (4,\ \nicefrac{3}{2},\ \nicefrac{1}{2},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ 1,\ 4,\ \nicefrac{3}{2},\ \nicefrac{1}{2},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ \nicefrac{1}{4},\ \nicefrac{1}{2},\ \nicefrac{1}{2})
\end{aligned}
$$

Figure 27: Two note sequences generated by the Markov model described above.

Finally the note sequences can be compared to find out which one is more probable according to the Markov model.

$$
\begin{aligned}
\bar{P}(Q^p_1) &= \nicefrac{1}{19}\big(\ \nicefrac{3}{19} + \nicefrac{1}{6} + \nicefrac{1}{6} + 1 + \nicefrac{1}{3} + \nicefrac{1}{6} + \nicefrac{2}{3} + \nicefrac{1}{3} + \nicefrac{1}{6} + \nicefrac{1}{6} + \nicefrac{1}{6} + 1 + \nicefrac{1}{3} +\\
&\qquad 1 + \nicefrac{1}{3} + 1 + \nicefrac{1}{3} + 1 + \nicefrac{1}{3}\big)\\
&= \nicefrac{503}{1083} \approx 0.464
\end{aligned}
$$

$$
\begin{aligned}
\bar{P}(Q^d_1) &= \nicefrac{1}{19}\big(\ \nicefrac{3}{38} + \nicefrac{1}{2} + \nicefrac{1}{2} + \nicefrac{1}{2} + \nicefrac{2}{3} + \nicefrac{2}{3} + \nicefrac{2}{3} + \nicefrac{1}{12} + \nicefrac{1}{3} + \nicefrac{1}{3} + \nicefrac{1}{2} + \nicefrac{1}{2} +\\
&\qquad \nicefrac{1}{2} + \nicefrac{1}{4} + \nicefrac{1}{2} + \nicefrac{1}{2} + 1 + \nicefrac{1}{6} + \nicefrac{1}{2}\big)\\
&= \nicefrac{997}{2166} \approx 0.46
\end{aligned}
$$

$$
\begin{aligned}
\bar{P}(Q^p_2) &= \nicefrac{1}{17}\big(\ \nicefrac{3}{19} + \nicefrac{1}{6} + \nicefrac{1}{6} + 1 + \nicefrac{1}{3} + \nicefrac{1}{2} + \nicefrac{1}{2} + \nicefrac{1}{2} + \nicefrac{1}{2} + \nicefrac{1}{2} + \nicefrac{1}{6} + \nicefrac{1}{4} + \nicefrac{1}{6} +\\
&\qquad \nicefrac{1}{2} + \nicefrac{1}{6} + \nicefrac{1}{3} + \nicefrac{1}{6}\big)\\
&= \nicefrac{1385}{3876} \approx 0.357
\end{aligned}
$$

$$
\begin{aligned}
\bar{P}(Q^d_2) &= \nicefrac{1}{19}\big(\ \nicefrac{1}{19} + \nicefrac{1}{2} + 1 + \nicefrac{1}{3} + \nicefrac{1}{2} + \nicefrac{1}{4} + \nicefrac{1}{2} + \nicefrac{1}{2} + 1 + \nicefrac{1}{3} + \nicefrac{1}{2} + \nicefrac{1}{2} + \nicefrac{1}{2} +\\
&\qquad \nicefrac{1}{2} + \nicefrac{1}{2} + \nicefrac{1}{4} + \nicefrac{1}{3}\big)\\
&= \nicefrac{9}{19} \approx 0.474
\end{aligned}
$$

Since $\bar{P}(Q_1^p)\bar{P}(Q_1^d) > \bar{P}(Q_2^p)\bar{P}(Q_2^d)$ the first sequence can be considered more close to the beginning of *Havona* according to the Markov model.

In the project described here some tweaks are incorporated for further improving the performance of Markov models: Two different *Smoothing*-techniques as well as *Chord-based Markov chains*.

Smoothing refers to techniques to counteract the zero-frequency problem. That means according to an oth-order Markov model, any unseen sequence of length o has the probability of 0. Indifferent of a sequence that is quite probable at first glance, according to model, if there is any unseen subsequence, the whole sequence has the probability of 0, because probabilities will be multiplied.

The first technique is *linear interpolation smoothing*: Lower-order Markov chains tend to consider few sequences as 0-probable, but are not very meaningful. Higher-order Markov chains are more meaningful, but tend to consider more sequences 0-probable. So one idea is to combine several Markov models up to some order bound O by combining their probabilities.[70] If P_o denotes the oth-order Markov model with its individual transition- and initial-probabilities A_o and Π_o, the smoothed Markov model can be described:

$$\hat{P}_{\text{Interpolation-smooth}}(q_t|q_{t-1}\dots q_0) = \sum_{o=1}^{O} \lambda_o P_o(q_t|q_{t-1}\dots q_0) \tag{23}$$

The scaling factors λ_o have to be chosen so that $\sum_{o=1}^{O} \lambda_o = 1$. In our project $\lambda_o = {}^1\!/O$ meaning an averaging of several Markov models. Beside of counteracting the zero-frequency problem, a local and a more global view on the music is combined.

Nevertheless, if the sequence contains an event that is not in the training set for the Markov model at all, the overall probability is still 0, independent of how probable the rest of the sequence is. *Additive smoothing*[71] rids the zero-frequency problem

70 See *Artificial Intelligence*, pp. 846 et seq. for a general depiction and *Markov Constraints*, p. 160 for one referring to Markov model music generation. For a more theoretical and rigorous discussion see Frederick Jelinek/Robert L. Mercer: *Interpolated Estimation of Markov Source Parameters from Sparse Data*, in: *Pattern Recognition in Practice. Proceedings of an International Workshop held in Amsterdam* 1980, pp. 381–397.

71 See Stanley F. Chen/Joshua Goodman: *An Empirical Study of Smoothing Techniques for Language Modeling*, in: *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics* 1996, pp. 310–318 for a comparison of different smoothing techniques. There on p. 311 it is argued, that additive smoothing generally performs poorly, but note in the case of this project it is used in combination with interpolation smoothing (related to what is there called Jelinek-Mercer-Smoothing).

of the model by assuming there is always a small probability for seeing yet unseen events. A generalized version of this problem can be stated as follows:

$$\hat{P}_{\text{Additive-smooth}}(q_t|q_{t-1}\dots q_0) = \frac{P(q_t|q_{t-1}\dots q_0)+\alpha}{1+\alpha N^{O+1}} \tag{24}$$

Where $\alpha > 0$ is a smoothing factor and N^{O+1} is the number of transition possibilities for a fixed-order Markov model.

Chord-based Markov chains means that several Markov chains are trained for each chord-type, defined by its pitch classes when the chord is transposed to C. Of course the events belonging to this chord are also transposed by the same amount, when training or when classifying with such a Markov model. Obviously within a major chord the major third might be expected to be more probable than within a minor chord, so the benefit of such a model in music is apparent. See figure 5 on p. 12 for an overview of the chord types to be found in the corpora of this project. So when 25 chord types are to be found, there is a separation between pitch and duration and order 0 up to 4 is involved, $2 \cdot 25 \cdot 5 = 250$ different Markov chains would be involved for each bassist.[72]

For applying chord based Markov chains the note succession to be classified should show the same chords like in the training corpus. The following table provides information concerning the relative amount of the chord types with respect to the duration they occupy within the training corpus. Chords with higher percentages are preferable when trying to classify a note succession.

$C_{pitchclasses}$	amount	$C_{pitchclasses}$	amount	$C_{pitchclasses}$	amount
$\varnothing$	35.02%	$\{0\ 4\ 7\ t\}$	18.27%	$\{0\ 3\ 7\ t\}$	15.89%
$\{0\ 4\ 7\}$	8.66%	$\{0\ 4\ 7\ e\}$	5.03%	$\{0\ 4\ 6\ e\}$	3.59%
$\{0\ 3\ 7\}$	2.78%	$\{0\ 3\ 4\ 7\ t\}$	2.24%	$\{0\ 2\ 4\ 7\}$	2.15%
$\{0\ 1\ 3\ 4\ 7\ t\}$	1.08%	$\{0\ 3\ 4\ 7\ e\}$	0.81%	$\{0\ 3\ 4\ 8\ t\}$	0.72%
$\{0\ 3\ 5\ 7\ t\}$	0.72%	$\{0\ 3\ 6\}$	0.58%	$\{0\ 3\ 6\ t\}$	0.54%
$\{0\ 7\}$	0.36%	$\{0\ 2\ 3\ 4\ 7\ t\}$	0.36%	$\{0\ 2\ 7\}$	0.36%
$\{0\ 1\ 4\ 6\ t\}$	0.31%	$\{0\ 1\ 3\ 7\ t\}$	0.18%	$\{0\ 1\ 4\ 7\ t\}$	0.18%
$\{0\ 2\ 4\ 7\ t\}$	0.09%	$\{0\ 3\ 5\ t\}$	0.07%		

Figure 28: Relative amount of chord types to be found in the Pastorius corpus.

[72] In fact there aren't because for Pastorius the chords $\{0\ \ 4\ \ 5\ \ 6\}$ and $\{0\ \ 7\ \ t\}$ are not to be found in the corpus and for Wooten only 10 chords are to be found, so $230 + 100 = 330$ overall Markov models are used for this thesis.

2.5 Classification results

As already explained in section 2.3.3 on p. 31 we estimate the performance of the models with a stratified 3-fold cross-validation. In the following confusion matrices, always the mean and standard deviation of the relative classification rates on the hold out test set are given. Although relative details are unusual, I prefer it for comparison between the different window sizes. See figure 3 on p. 10 for the absolute number of training items. Since used in a 3-fold cross-validation the relative details in the matrices correspond to the size of the test sets, i.e. $^1/_3$ of the overall numbers, while being trained on the other $^2/_3$. ACC stands for accuracy and gives the number of correctly classified items relative to the overall number of items in the test set.

In figure 29 on p. 42 it can be seen that the overall classification results are quite decent, always exceeding 90%. So for classifying whole pieces, each consisting of dozens of windows, the majority vote among all windows would probably result in correct classification of the pieces. Nevertheless, the amount of correctly classified Wooten is always higher. A reason for this could be the more repetitive style of playing of Wooten resulting in features that are more similar to each other.

For the sake of comparability the same windowing was made for the following confusion matrices of the Markov chain classification in figure 30 on p. 43. Note that in contrast to the feature classification described above, here windowing isn't an inherent procedure. It's more natural to learn on sequences without windows because a sequence of N notes contribute with $N - O$ events to the transition probabilities of an Oth-order Markov chain. So when splitting the sequence into windows, less notes contribute to transition probabilities. This is partly compensated by overlapping windows, but they don't fully make up for it when the average note duration isn't sufficiently small, i.e. there are relatively few notes per window. That is one reason why increasing the window size in principle increases the performance of the Markov classification.[73] For the following confusion matrices, both smoothing techniques described in section 2.4 are used, for orders 0 to 4 and with the additive term $\alpha = 0.001$. Also note, that there are cases of *undecided* estimates – windows for which the Markov probabilities of being Pastorius or Wooten are equal.

[73] Apart from greatly varying performance of classifying Wooten across the folds in the case of 4ql.

win. size = 2ql		Actual: Pastorius	Actual: Wooten
Estimate	Pastorius	88.11% (±1.31%)	4.5% (±0.52%)
	Wooten	11.89% (±1.31%)	95.5% (±0.52%)
	ACC = 92.82% (±0.44%)		

win. size = 4ql		Actual: Pastorius	Actual: Wooten
Estimate	Pastorius	88.38% (±1.29%)	5.84% (±0.66%)
	Wooten	11.62% (±1.29%)	94.16% (±0.66%)
	ACC = 92.01% (±0.13%)		

win. size = 6ql		Actual: Pastorius	Actual: Wooten
Estimate	Pastorius	87.95% (±1.44%)	4.7% (±0.63%)
	Wooten	12.05% (±1.44%)	95.3% (±0.63%)
	ACC = 92.52% (±0.71%)		

win. size = 8ql		Actual: Pastorius	Actual: Wooten
Estimate	Pastorius	87.82% (±1.24%)	5.87% (±1.42%)
	Wooten	12.18% (±1.24%)	94.13% (±1.42%)
	ACC = 91.74% (±0.48%)		

win. size = 10ql		Actual: Pastorius	Actual: Wooten
Estimate	Pastorius	85.91% (±0.89%)	3.95% (±0.7%)
	Wooten	14.09% (±0.89%)	96.05% (±0.7%)
	ACC = 92.21% (±0.38%)		

win. size = 12ql		Actual: Pastorius	Actual: Wooten
Estimate	Pastorius	86.3% (±5.01%)	7.01% (±1.01%)
	Wooten	13.7% (±5.01%)	92.99% (±1.01%)
	ACC = 90.47% (±1.38%)		

Figure 29: Confusion matrices for Gradient Tree Boosting

win. size = 2ql		Actual: Pastorius	Actual: Wooten
Estimate	**Pastorius**	91.43% (±0.79%)	17.71% (±0.98%)
	Wooten	7.61% (±1.03%)	82.27% (±0.97%)
	Undecided	0.96% (±0.25%)	0.03% (±0.02%)
ACC = 85.59% (±0.88%)			

win. size = 4ql		Actual: Pastorius	Actual: Wooten
Estimate	**Pastorius**	94.24% (±2.46%)	26.45% (±25.03%)
	Wooten	4.95% (±2.52%)	73.55% (±25.03%)
	Undecided	0.81% (±0.18%)	0% (±0%)
ACC = 81.26% (±14.78%)			

win. size = 6ql		Actual: Pastorius	Actual: Wooten
Estimate	**Pastorius**	91.99% (±1.63%)	6.35% (±0.18%)
	Wooten	7.8% (±1.58%)	93.65% (±0.18%)
	Undecided	0.21% (±0.17%)	0% (±0%)
ACC = 93.02% (±0.64%)			

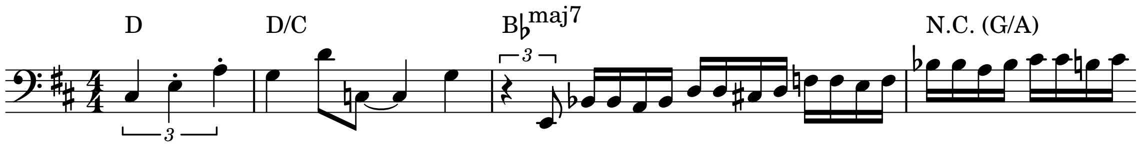
win. size = 8ql		Actual: Pastorius	Actual: Wooten
Estimate	**Pastorius**	93.64% (±1.35%)	6.6% (±0.6%)
	Wooten	5.81% (±0.98%)	93.4% (±0.6%)
	Undecided	0.55% (±0.6%)	0% (±0%)
ACC = 93.49% (±0.6%)			

win. size = 10ql		Actual: Pastorius	Actual: Wooten
Estimate	**Pastorius**	94% (±0.17%)	5.15% (±0.69%)
	Wooten	5.89% (±0.01%)	94.85% (±0.69%)
	Undecided	0.12% (±0.16%)	0% (±0%)
ACC = 94.53% (±0.37%)			

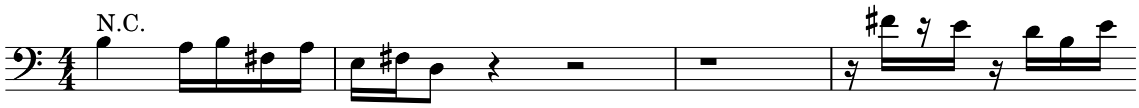
win. size = 12ql		Actual: Pastorius	Actual: Wooten
Estimate	**Pastorius**	90.78% (±2.07%)	4.56% (±1.9%)
	Wooten	9.22% (±2.07%)	95.44% (±1.9%)
	Undecided	0% (±0%)	0% (±0%)
ACC = 93.68% (±1.62%)			

Figure 30: Confusion matrices for chord based Markov models

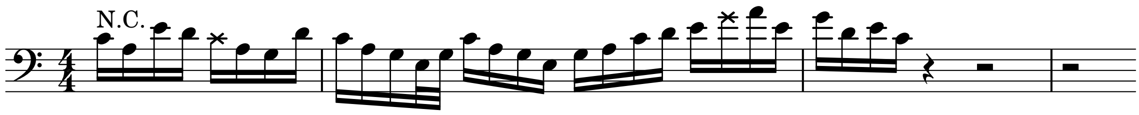
See figure 31 for three exemplary windows for both bassists that have been misclassified in the 12ql-case in both procedures. 12ql doesn't give the best performance in both cases, but long windows are better for human evaluation. All windows are not completely incomprehensible: 31a and 31c show long and rather diatonic 16th runs, what could be considered more typical for Wooten. 31b is nearly occupied by rests for $^2/_3$, so being unsure in the case of few notes is also understandable. 31d and 31f show some amount of chromaticism and 31f also shows some non-chord notes, e.g. during $G\flat^7$ 3 of 4 notes are non-chord notes. Both chromaticism and non-chord notes can be considered more typical for Pastorius. 31e is an example where the restriction to monophonic music could exhibit its drawback. Only the upper voice is taken into account and the lower one is ignored. So when regarding a "filtered" version of a window, a misclassification isn't a surprise either.



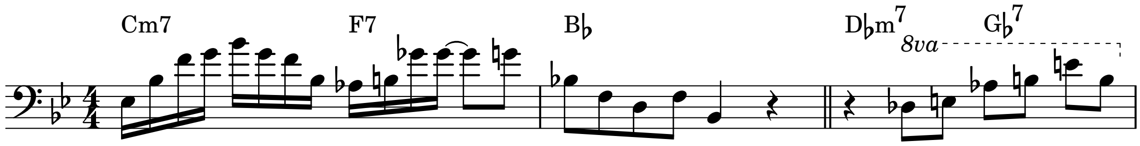
a: from Pastorius' *Bright Size Life*



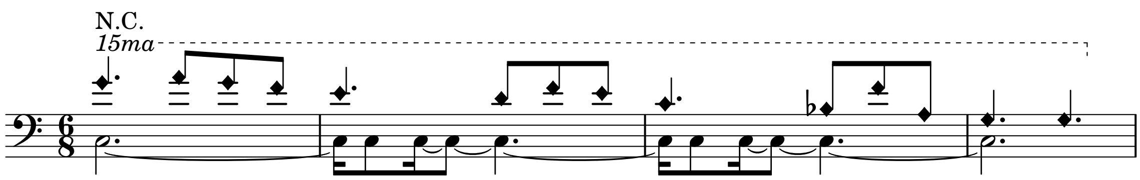
b: from Pastorius' *Havona*



c: from Pastorius' *Port Of Entry*



d: from Wooten's *Cherokee (Indian Love Song)*



e: from Wooten's *Norwegian Wood (This Bird Has Flown)*

f: from Wooten's *Sinister Minister*

Figure 31: Some 12ql windows that have been misclassified with both the Gradient Tree Boosting and the Markov classification. Note that in subfigure e the octaving *15ma* refers to the upper voice only. The lower voice isn't used in this project anyways because it is restricted to monophonic music.

3 Modification

3.1 Overview

The classification procedures described in the previous chapters will now be utilized for a modification process. An existing note succession with a corresponding chord annotation should be modified in such a way that it comes closer to Pastorius according to the classification. In that sense, it can be considered a style changing procedure.

this task is viewed as a local search for a multi-objective optimization.[74] The main idea is to start from a note succession and try out “neighbors”. If a neighbor is better than the original one, according to an objective function, the neighbor is saved and the process is iteratively continued with this one. What is considered a neighbor to a note succession is yet to be defined, see section 3.2. The objective function is responsible for increasing the probability of a given piece of music to be classified as Jaco Pastorius. Beside the two classification procedures described in the previous sections, there are other objectives as well, described in section 3.3.

So there isn’t a single objective function, but several ones. Since in principle it is possible that for a given neighbor the value of one objective function increases and another one decreases, it is not quite clear if the neighbor should be considered “better”. In this context the term of Pareto optimality[75] is important: a neighbor is regarded better if and only if the values of all objective functions are equal or higher and at least one value is higher. This is a strict regulation that ensures that the note succession doesn’t become less similar to Pastorius in any regard.

Most metaheuristics comprise of the idea of iteratively doing small random changes so that the value of the objective function becomes better. Since this procedure is most probably threatened to be trapped in a local optimum, many efforts in metaheuristics are employed to overcome local optima for finding the global one or at least solutions better than the first local optimum: For example genetic algorithms have been applied for melody generation.[76] But there is a relaxing property in the

[74] For a thorough but still accessible overview of stochastic optimization see Sean Luke: *Essentials of Metaheuristics*, 2nd ed., Raleigh 2013, available at `http://cs.gmu.edu/~sean/book/metaheuristics` (henceforth cited as *Metaheuristics*).

[75] Ibid., pp. 133 et seqq.

[76] See John A. Biles: *GenJam: A Genetic Algorithm for Generating Jazz Solos*, in: *Proceedings of the 1994 International Computer Music Conference*, 1994, pp. 131–137 and George Papadopoulos/Geraint Wiggins: *A Genetic Algorithm for the Generation of Jazz Melodies*, in: *Proceedings of the Finnish Conference on Artificial Intelligence*, 1998, pp. 7–9.

use case of this thesis: There is no need for finding the global optimum anyway. The idea is to change a bass line so that it becomes closer to Pastorius – but it should still be recognized as the original bass line. Finding the global optimum would mean to allow to completely throw away the original one and replace it by the "most-Pastorius-alike" bass line ever possible. That one would probably be one of the Pastorius corpus or a combination of the most frequent phrases from it. In our case it is more desirable not to throw away the original bass line, but to change it until there are no more small modifications to improve the objective functions. So finding a local optimum is perfectly fine.

Figure 32 shows a decision surface of only two successive pitches regarding the average smoothed Pastorius Markov probability with order 0 and 1. Note how rugged it already is in this simple case. It is obvious that a simple hill climbing approach wouldn't find the global optimum in most cases, because there are multiple local maxima. In the scope of this thesis, this is a preferable property because it will prevent the pitches being changed too strongly.

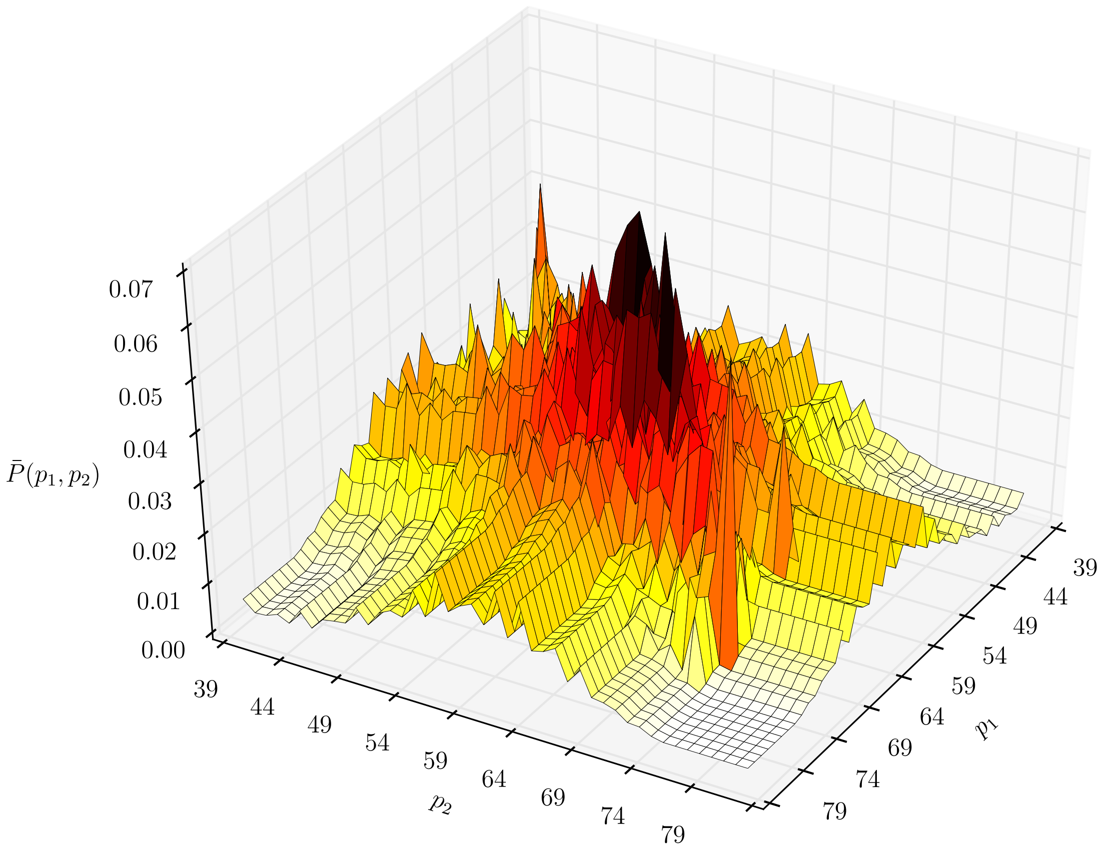


Figure 32: Surface of an objective function of two successive pitches regarding the average smoothed Pastorius Markov probability (order 0 and 1).

But even if there is no need to overcome local optima, there is still something in the toolbox of metaheuristics that can be deployed. Tabu search[77] helps to accelerate the search. That means a history of already tried out neighbors is saved, so that they won't be tried out again. In its original form the oldest item from the tabu history is removed if the history exceeds some size limit. In this project it is implemented in such a way, that the history is emptied after a candidate was accepted. From that on, the tabu history is built anew. This allows the tabu list to not contain complete neighbors, but only possible modifications to the current state.

3.2 Neighborhood

The neighborhood is a rather problematic field of the optimization process. Ideally one wants to have a manageable amount of neighbors so that every one can be tried to find out which one is best. Unfortunately the amount isn't manageable in the case of this thesis. Below the four different possible modifications to a given note succession are described. Note that only a single modification can be applied for achieving a neighbor.

- Changing the pitch of a single note by any semitone between a major third downwards and a major third upwards.

A major third is an amount that seems a reasonable balance between too small and too large. A second may be too small, e.g. because in a triadic arpeggio it is more common to continue with a third instead of of a second. So if we have N notes in the note succession, here there are $8N$ possibilities for a neighbor.

- Changing the duration of two notes so that the overall duration of the note succession stays the same.

The duration of a note can be changed to any duration to be found in the corpus of Pastorius – that are 17 different ones – excluding the duration that is the current one. Then another note has to be found that can be changed, so that the overall duration isn't changed. So for this second note there is no choice in duration. In practice not all note positions are possible, e.g. because the alteration of the duration of the second note is fixed (because the overall duration has to stay the same) and that duration could possibly not be in the durations to be found in the corpus of Pastorius. But disregarding this fact, at maximum, here there are $16N(N-1)$ possibilities for a neighbor.

[77] *Metaheuristics*, pp. 26 et seqq.

- A note is divided into multiple ones.

The duration of a note can be divided into two ones having the duration ($^1/_2$, $^1/_2$) or ($^2/_3$, $^1/_3$) or into three notes having the duration ($^1/_3$, $^1/_3$, $^1/_3$) relative to the original duration. The first pitch of the new notes is the pitch of the original note and the remaining pitches are decided by the Markov model. When disregarding, that the Markov model decides in a weighted random manner and thus can produce different pitches each time tried, here there are still $3N$ possibilities for a neighbor.

- Two notes are joined into a single one.

Two nearby notes are joined into a single note with the duration being the sum of both original ones. The pitch of the new note is one of the pitches of the two original ones, determined in a weighted random manner with the weights being proportional to the note durations. Again, let's ignore the pitch-related possibilities, so there are still $N - 1$ possibilities in this case.

When summing all possibilities, we are left with $16N^2 - 4N - 1$ neighbors. Since this amount depends on the number of notes which can vary during the optimization process (by joining or dividing notes), the amount of possible neighbors isn't stable during the optimization. Although this amount isn't exponential, there are too many neighbors to try all of them. So one has to be satisfied with randomly picking one possible neighbor. If this one isn't better than the current state, one tries another one, etc.

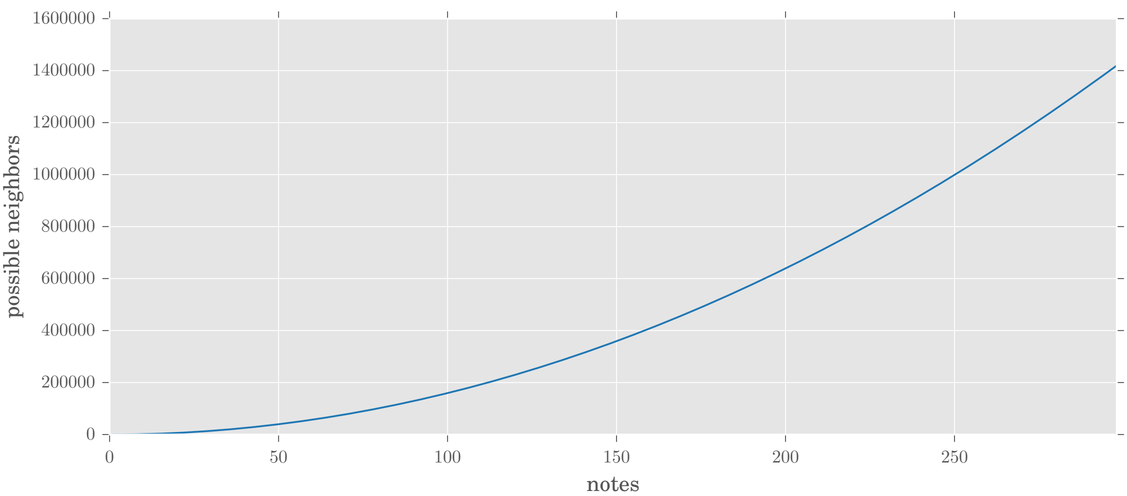


Figure 33: Number of possible neighbors, dependent on the number of notes of the sequence.

One could think of how to incorporate the Markov models more strongly, to try less possibilities by just using pitches and durations that are probable according to the

Markov model. By this means one could speed up the optimization, but there would be quite a strong emphasis on the Markov model. With the current rather random neighborhood, the Markov model and the other objectives are equally responsible for the final result.

3.3 Objectives

There are four objectives that are considered during the optimization process. Here each of them is explained.

- Feature classification

This corresponds to sections 2.2 and 2.3. In short: The note succession is divided into windows by a given size (with hop size 50%), features are extracted from each of them and then each of them is being classified by the Gradient Tree Boosting algorithm. Instead of using a binary decision, the logarithmic probability of being Pastorius is used. The sum of them forms the value of this first objective.

- Markov classification

The value of this objective is the average smoothed Markov probability of Pastorius, described in section 2.4.

- Ratio of the Pastorius and Wooten Markov property

The preceding objective only takes the Pastorius Markov model into account. So it also rewards changes that make a given note succession more close to general musical characteristics or fusion bass guitar characteristics. To foster the specific characteristics of Pastorius, the value of this objective is the ratio of the average smoothed Markov probability of Pastorius and average smoothed Markov probability of Wooten.

To get a sense of how different this is from the previous objective, see here the surface of this objective regarding the succession of two pitches. Compare this with figure 32 on p. 46.

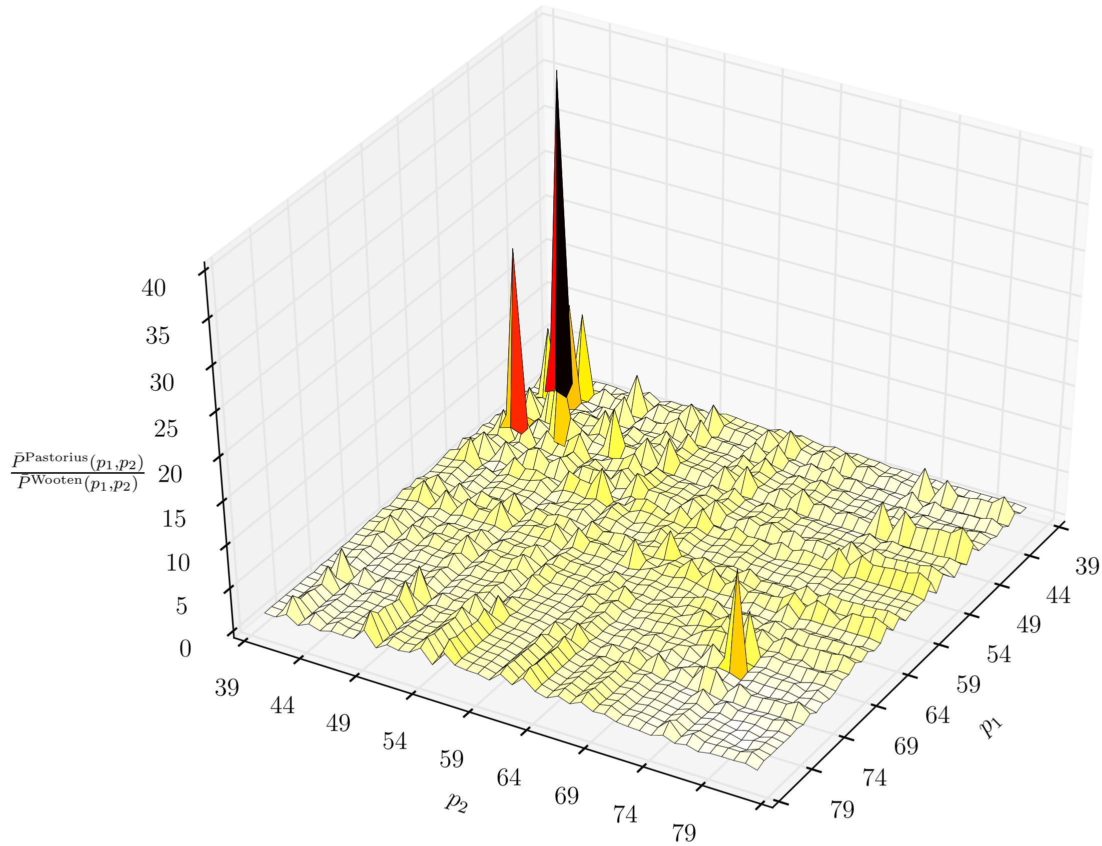


Figure 34: Surface of an objective function of two successive pitches regarding the ratio of the average smoothed Pastorius Markov probability and the average smoothed Wooten Markov probability (order 0 and 1).

- Time correlations for chord repetitions

This is an objective not related to the ones yet described. The main idea is to capture some large scale structural similarities with the pieces of Pastorius. A shallow idea of large scale structure should also be given by the Gradient Tree Boosting, because it also considers the relative position of the window. But in practice large scale structure is something one misses most in the generated music. So this is an additional approach to include that.

The assumption is, that structure evolves by the absence or presence of repetition. Repetition in a "closed" view is already captured[78] – but a more "global" view should be modeled here, e.g. the varied reoccurrence of material already played some time ago. The second assumption is that such kind of repetition most probably occurs when the relative changes in harmony also repeat. See figure 35 for an example:

[78] See section 2.2.2.6 on page 14.

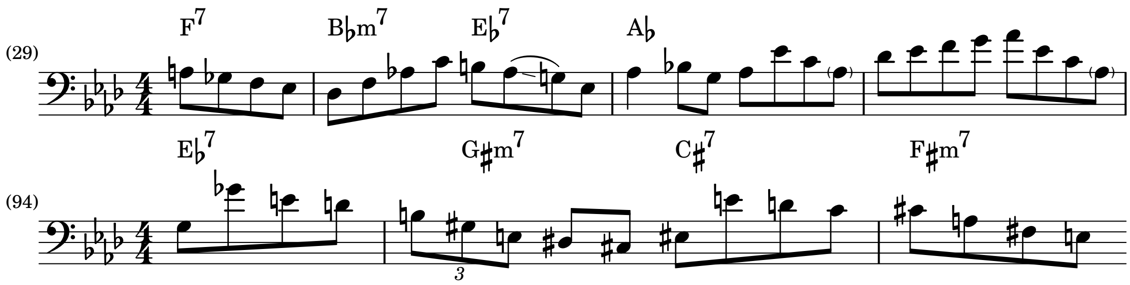


Figure 35: Two parts of Pastorius' *Donna Lee* with the same intervals between the roots of the chord progression (when enharmonic change is ignored): ascending fourths.

When enharmonic change is ignored, the roots of the chords form a progression of ascending fourths in both parts. Although there isn't a real theme or motif that is repeated, the same kind of material is used – the mixture of scale sections and triadic sections are at least kind of related. The two notes sequences can be transformed into a piano-roll representation:

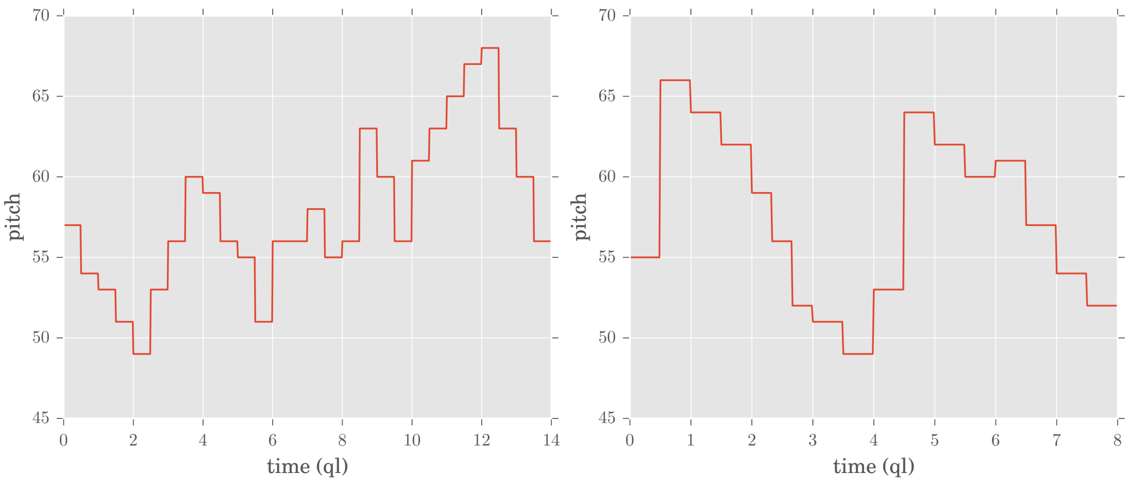


Figure 36: Plot of piano roll representation of the music from figure 35.

After subtracting the mean and dividing by 64 from these piano-roll representations[79], a circular cross-correlation is performed, as explained in section 2.2.2.6 on p. 14.

[79] Since midi pitches range between 0 and 127, by doing this it is ensured that it is scaled within -1 and 1.

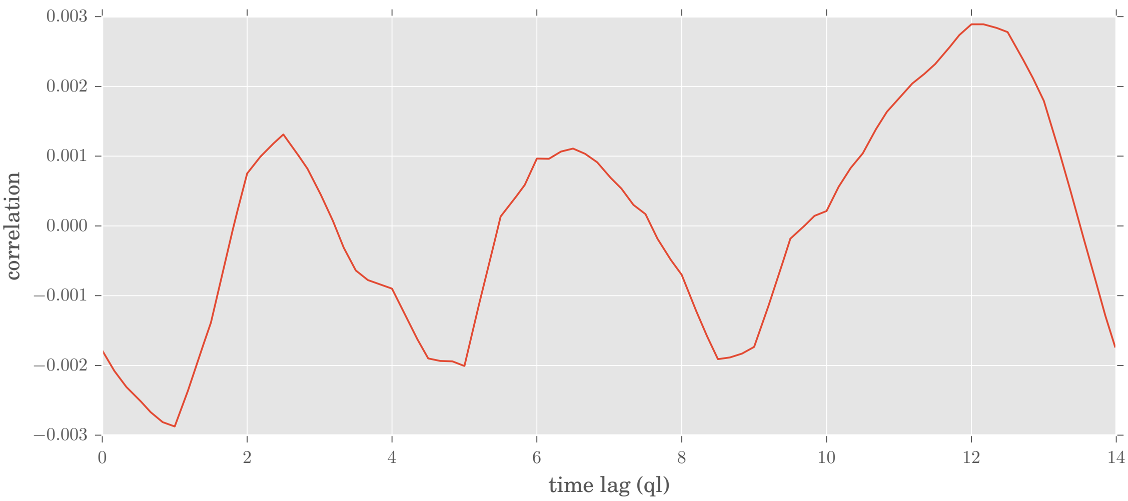


Figure 37: Circular correlations of the piano-roll representations of the music from figure 35.

Here one sees three prominent maxima. These correspond to rough structural similarities of the second note sequence with the first. Let's consider the greatest maximum with a lag of 12 ql.

Figure 38: Two parts of figure 35, put together, with a lag of 12ql and circular wrapping in the second part.

Figure 38 shows the examples from figure 35 with a lag of 12ql for the second example. Since it's circular correlation, the end of the second example wraps at the beginning again. In this case there is the maximum "pseudo-parallel" movement of both examples. Even although that is not directly perceptible in music, the assumption in this thesis is, that it contributes to the large scale feeling of coherence within a piece of music.

And that's how it is applied in the optimization: All cross-correlations between parts with related chord-progressions in the Pastorius corpus are pre-computed. During the optimization also the cross-correlations between parts with related chord-progressions are computed. Then the inner product between each correlation of the piece of music to be optimized and the ones of the Pastorius corpus are computed and the maximum one is returned as value of this objective. Since the dot product of correlations of different lengths cannot be computed, all correlations of the same progression length are brought to the same length by linear interpolation.

By that means one ensures that the correlation within the chord progressions becomes more similar to one of the examples in the Pastorius corpus.

3.4 Results

3.4.1 Ex. 1: *New Britain* (traditional)

In contrast to the classification, where the performance can be evaluated relatively impartial, such an objective evaluation is not possible for the modification. One can only try it out and evaluate the result subjectively. So the assessment heavily depends on the judges, their musical knowledge, taste, etc.

As a first example, a simple traditional melody has been chosen: *New Britain* (earliest sources 1831), often sung along with the Christian hymn *Amazing Grace*.[80] See here its original version:[81]

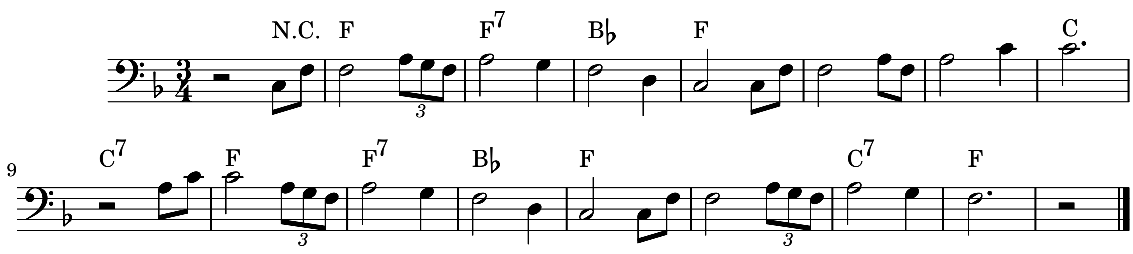


Figure 39: *New Britain* resp. *Amazing Grace*, original version.

During the modification process, 3398 different changes have been tried out whereby only 14 ones have been accepted by Pareto optimality. See the final version below:

[80] Sheet music origins from `http://www.free-notes.net/cgi-bin/noten_Song.pl?song=Amazing+Grace`, arrangement by the author.

[81] *Original* here means that it is original for the modification process. Of course the arrangement isn't original.

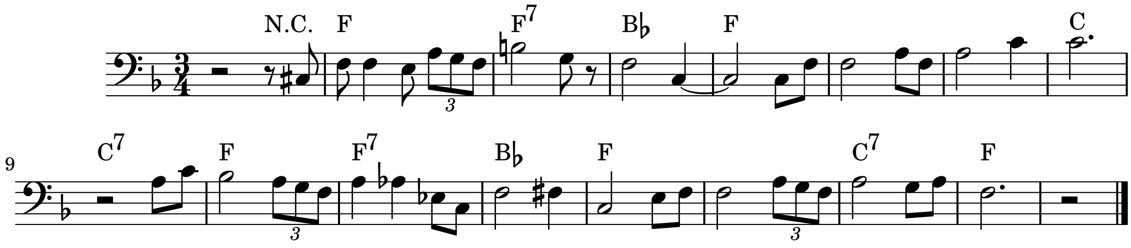


Figure 40: *New Britain* resp. *Amazing Grace*, "jaconized" version.

It is obvious that major parts of the original version remain intact. Here is a list of the accepted changes:[82]

1. B. 2,1: half note f was split into two quarter notes with pitches $\begin{pmatrix} e & f \end{pmatrix}$.
2. B. 11,3: g changed to $e\flat$
3. B. 1,3; c changed to $c\sharp$.
4. B. 11,3: quarter note $e\flat$ was split into two eighths with pitch $\begin{pmatrix} e\flat & c \end{pmatrix}$.
5. B. 15,3: quarter note g was split into two eighths with pitch $\begin{pmatrix} g & a \end{pmatrix}$.
6. B. 4,3: quarter note d and b. 5,1: half note c have been joined into a single one with pitch c.
7. B. 13,3: c changed to e.
8. B. 12,3: d changed to $e\flat$.
9. B. 3,3: quarter note g was split into two eighths with pitches $\begin{pmatrix} g & \rho \end{pmatrix}$.
10. B. 11,1: half note a was split into two quarter notes with pitches $\begin{pmatrix} a & g\sharp \end{pmatrix}$.
11. B. 10,1: c changed to $b\flat$.
12. B. 2,1: 2nd quarter note e was changed to an eighth and b. 1,1: duration of half note rest was changed to $^5/_2$.
13. B. 12,3: $e\flat$ changed to $f\sharp$.
14. B. 3,1: a changed to b.

[82] For the sake of readability the list uses pitch class names instead of numbers. ρ stands for rest. B. stands for bar and the number following the comma indicates the beat. So B. 1,3 means the 3rd beat of the 1st bar.

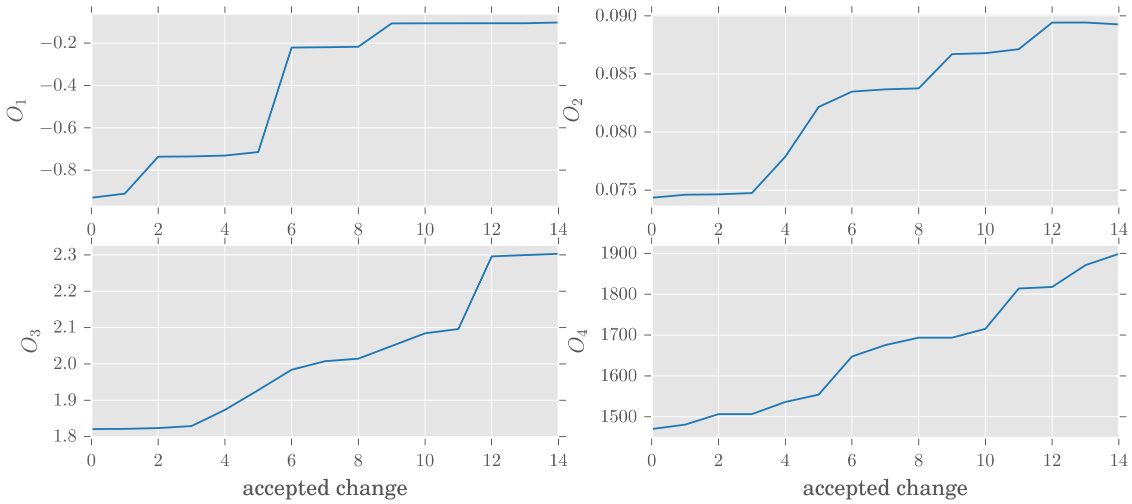


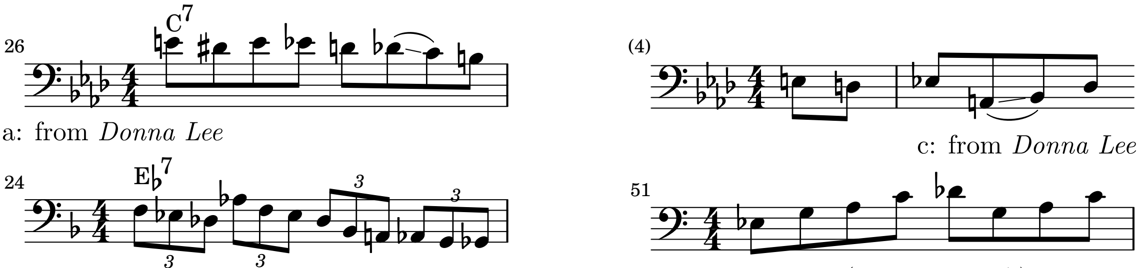
Figure 41: Evolvement of the four objectives during the optimization process for *New Britain*. O_1, O_2, O_3 and O_4 are ordered in the same way the objectives are described in section 3.3.[83]

From the author's view it is hard to evaluate the final result aesthetically since he is quite biased. The tieing of notes from bars 4 to 5 seems to interrupt the arc of suspense, thus it rather seems to be a mistake from a musical viewpoint. The run from bar 9 seems interesting and coherent. The two eighths in bar 9 as well as the ones in bar 15 seem to surround the following principal note – especially the first example can be considered a *Doppelvorschlag*[84] –, which lets the music appear quite natural. Some succession seem harsh, like the succession of the minor to the major third of the chord in bar 11 or the melodic succession of a semitone and a tritone in bars 12–13, but such harsh elements are not unusual in the music of Pastorius, see figure 42. Maybe they seem spurious, since one rather bears in mind the consonant original version, which is still noticeable in the modified version to a big extend. But from that point of view, the result is a successful blending of the original version and

[83] For this and the following evolvement curves holds that in some cases they can be a bit inaccurate. During the optimization a list of pitches and durations is considered. Accepted steps are saved as MusicXML and from these accepted steps the curves are generated. So if e.g. the lists contain succession of rests, they may be joined in the MusicXML files. So the curves can differ to a certain degree from the actual values during the optimization. Nevertheless they give an impression of tendency and magnitude. By inaccuracies due to such differences between the lists and the MusicXML files e.g. the minor decrease in O_2 by ca. $1.63 \cdot 10^{-4}$ from change 13 to 14 can be explained.

[84] Untranslatable. See Frederick Neumann: *Ornamentation in Baroque and Post-baroque Music. With special emphasis on J.S. Bach*, 3rd ed., Princeton 1983, p. 488. According to Neumann the translation "double appoggiatura" would be a modern coinage. It means the sequence of lower neighbor, upper neighbor and principal note.

the style of Pastorius, even if it is very doubtful if Pastorius would have improvised over *New Britain* like this.



a: from *Donna Lee*

b: from *The Days of Wine and Roses*

c: from *Donna Lee*

d: from *(Used To Be A) Cha Cha*

Figure 42: Pastorius examples, that could be the model for "harsh" results in the modification. a and b: Succession of the minor on the major third of the chord. c and d: Melodic succession of a semitone and a tritone.

3.4.2 Ex. 2: Johannes Brahms' Cello Sonata No. 1 in E minor, Op. 38

After the previous, rather naive, traditional music example, a more sophisticated one follows: With the first phrase of the Cello Sonata No. 1 in E minor, Op. 38 (1862–65) by Johannes Brahms, the modification process is investigated within a very different musical context.

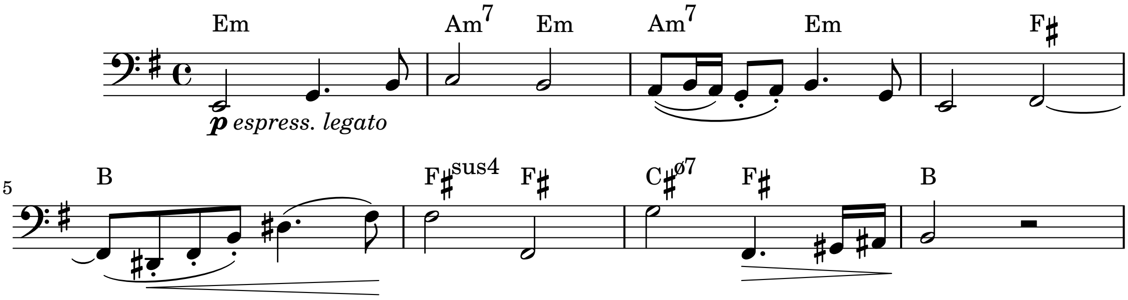


Figure 43: First bars of the opening movement of Johannes Brahms' Cello Sonata No. 1.[85]

During the modification process, 4103 different changes have been tried out whereby 27 ones have been accepted by Pareto optimality. See the final version below:

[85] Chord annotations by the author with regards to figure 28 on p. 40.

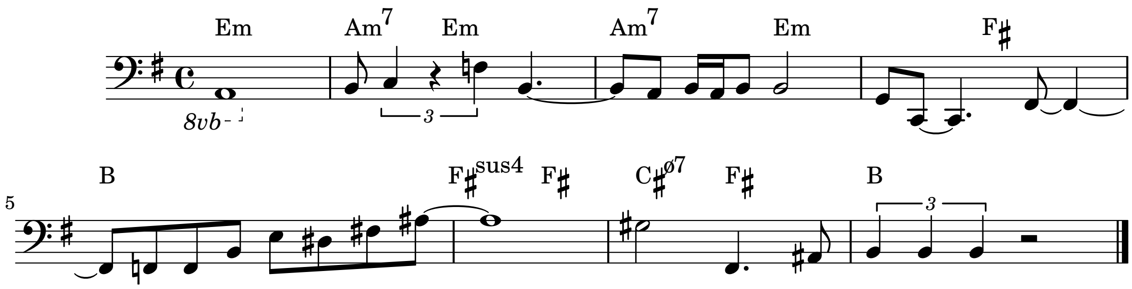


Figure 44: First bars of the opening movement of Johannes Brahms' Cello Sonata No. 1, "jaconized" version.

Again, see the list of accepted changes.

1. B. 1,1: half note e and b. 1,3: dotted quarter note g have been joined into a single one with pitch e.
2. B. 6,1: duration of tied ($5/2$) note $f\sharp$ was changed to half and b. 4,2: half note $f\sharp$ was changed to $5/2$.
3. B. 8,3: half rest was split into three quarter triplet rests. (Had no effect on final result.)
4. B. 8,1: half note b was split into three quarter triplet notes with pitch b.
5. B. 7,4: two sixteenths $\left(g\sharp \quad a\sharp\right)$ have been joined into a single one with pitch $a\sharp$.
6. B. 5,1: $d\sharp$ changed to f
7. B. 4,1: e changed to $c\sharp$
8. B. 7,1: g changed to $a\sharp$
9. B. 2,1: half note c was split into three quarter triplet notes with pitch $\left(c \quad \rho \quad f\right)$.
10. B. 6: two half notes $f\sharp$ have been joined into a single one with the same pitch.
11. B. 4,1: $c\sharp$ changed to b
12. B. 5,1: second eighth $f\sharp$ changed to f
13. B. 5,2: dotted quarter note $e\flat$ was split into a quarter and eighth with the same pitch.
14. B. 7,1: $a\sharp$ changed to $g\sharp$
15. B. 3,2: first eighth g changed to b
16. B. 5,1: second eighth f changed to e
17. B. 5,1: second eighth e changed to c
18. B. 4,1: half note b changed to $d\sharp$
19. B. 5,3: duration of quarter note $d\sharp$ was changed to eighth and b. 1,1: double dotted half note e was changed to whole note.
20. B. 3,3–4: eighth note a and dotted quarter note b have been joined into a single one with pitch b.
21. B. 5: last eighth $f\sharp$ and the following tied whole note, was changed to $a\sharp$
22. B. 1,1: whole note e was changed to $c\sharp$
23. B. 1,1: whole note $c\sharp$ was changed to a
24. B. 5,2: first eighth c was changed to $d\sharp$
25. B. 5,3: first eighth $d\sharp$ was changed to e
26. B. 5,2: first eighth $d\sharp$ was changed to f
27. B. 4,1: half note beginning at second eighth $d\sharp$ was changed to c

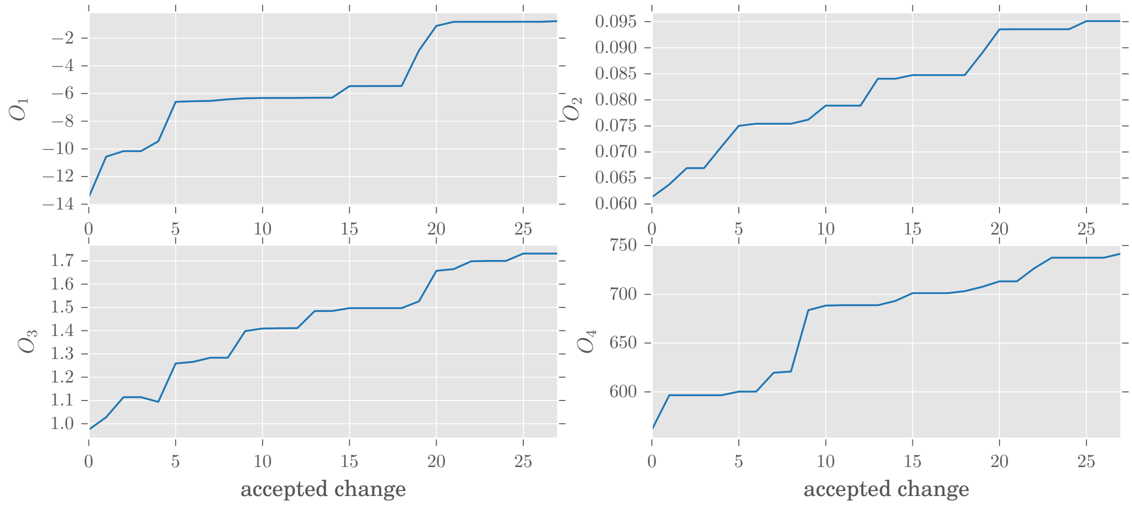


Figure 45: Evolvement of the four objectives during the optimization process for the Brahms example. O_1, O_2, O_3 and O_4 are ordered in the same way the objectives are described in section 3.3.

Again, despite the bias, some remarks on the result: The very low *a* in contra octave at the beginning and the triplet repeats of *b* at the ending are both kind of witty and act like brackets around the phrase. The second bar shows a rather a-metrical rhythmic structure, something that seems unfavorable in this way. The result may have been better without step 19 being accepted. The same musical unfavorability holds for bar 4, in a less critical way. So paying attention to the metrical structure is definitely something that could be improved in the modification process. The second half of bar 5 could be considered a mediant substitution, something quite common in jazz.[86] Ignoring the suspension of the fourth in the first half of bar 6 seems relatively adventurous, but note that in the corpus there is no suspended chord to be found,[87] so there isn't a model to be found in this case. Although one may regard the result as a blending of Pastorius into the Brahms excerpt, I find it less successful then the previous example.

3.4.3 Ex. 3: Brainsheep's *Shepherd's Tale*

As the last example the bass line of a rock song is considered. It is *Shepherd's Tale* (2011) of the band Brainsheep, of which the bass line was composed by Stefan Escaida. He made the transcription of his bass line, specifically for the experiment of the modification in the scope of this thesis.

[86] *Jazz Theory and Practice*, p. 111 et seqq.

[87] See figure 28 on p. 40.

Figure 46: Bass line of Brainsheep's *Shepherd's Tale*.

Since a Cm^9 chord doesn't have a model in the corpus[88], for the modification all Cm^9 has been changed into Cm^7. Although there are no suspended chords as well, these haven't been changed. During the modification process, 3210 different changes have been tried out whereby 94 ones have been accepted by Pareto optimality. See the final version below:

[88] See figure 28 on p. 40.

Figure 47: Bass line of Brainsheep's *Shepherd's Tale*, "jaconized" version.

Unlike the previous examples, this time the final result won't be described by the author. Instead a quote from the author of the original bass line, Stefan Escaida, should be given as evaluation:

> I think the result is interesting and certainly it has both the original component as well an additional flavour from Pastorius and jazz in general. Nonetheless, this statement is much clearer for the second set of chord changes, than for the first one. In the first section the revamped version doesn't sound very musical to my ears, i.e. very random, whereas the second version seems more natural and the style features are much more

recognizable. I imagine that a tool like this could be useful for someone who's already a playing musician interested in learning the "language" of a new musical style.

As far as I understand, the reinterpretation considers only melody and harmony factors, but from a bass players' point of view, the constraints due to a band's rhythm section should be equally important. It would be interesting to see how these constraints would eventually improve the result. Also, a factor that was handled implicitly was the timbre of the instrument. When listening to the result on a fretless bass sound-bank the resemblance to Pastorius was more evident.[89]

[89] Stefan Escaida, September 9, 2015.

4 Further Investigations

In this section a couple of additional, rather unrelated, small research findings, developed during this thesis project, shall be presented.

By his own accounts Jaco Pastorius has been strongly influenced by Charlie Parker,[90] so it seems natural, to try to find relations between Parker and Pastorius. For that purpose some Charlie Parker solos have been transcribed[91] for finding commonalities and quotations.[92]

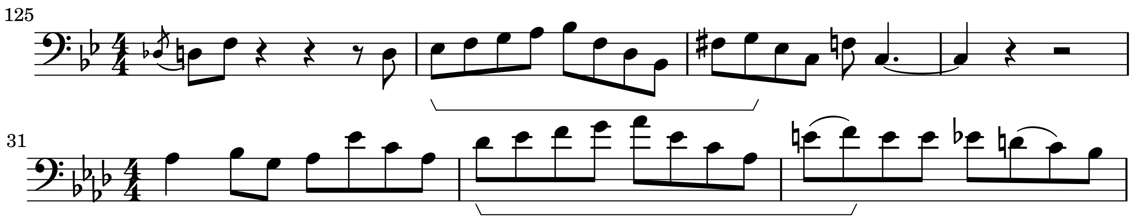


Figure 48: Bars 125–128 of Parker's *Anthropology* (above), bars 31–33 of Pastorius' version of *Donna Lee* (below)

The figure above shows a literal quote from Charlie Parker's *Anthropology* in Jaco Pastorius' version of *Donna Lee*. The majority is just an ascending scale and descending triad, but the final intervals (minor sixth and minor second) makes is recognizable. Note that although the author of *Donna Lee* probably is Charlie Parker,[93] this line isn't to be found in Parker's version of *Donna Lee*. So maybe it is an unconscious reference of Pastorius to Parker. Of course one can also search for retrograde or inversion quotes, but that seems very far from the musical sphere of both musicians. See figure 49 for an example.

---

[90] Steve Rosen: *Portrait of Jaco*, in: *International Musician and Recording World* August 1978, available online via `http://jacopastorius.com/features/interviews/portrait-of-jaco/`.

[91] See appendix E.4.

[92] For an overview of finding repetitions in symbolic music representations see Berit Janssen et al.: *Discovering repeated patterns in music: state of knowledge, challenges perspectives*, in: Mitsuko Aramaki et al. (eds.): *Sound and Muic. 10th International Symposium on Computer Music Multidisciplinary Research 2013*, 2014, pp. 277–297.

[93] It remains unclear to some extend if Miles Davis or Charlie Parker is the author. See *Jaco Portrait*, p. 6.

One can try to find something that could be considered a signature in the terminology of David Cope.[94] The table in figure 50 on p. 64 is about the interval succession $\begin{pmatrix}-1 & -1 & -1 & 4 & 3\end{pmatrix}$ with a tolerance of 2 semitones (i.e. one interval could differ by a whole tone or two intervals could differ by a semitone each). Although one rather could call it a motif of *Donna Lee*, it also can be found in some other works. But nevertheless one wouldn't consider it as a characteristic signature, like in the examples of Cope.

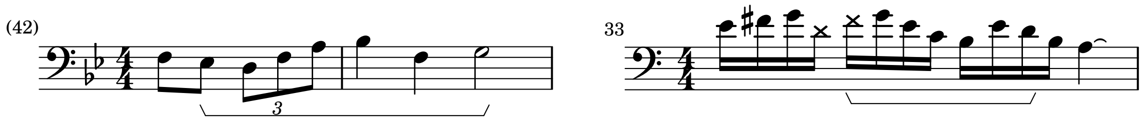


Figure 49: Bar 42–43 of Parker's *Moose the Mooche* (left), and bar 33 of Pastorius' *Havona* (right). Both share the interval succession in inverted relationship $\begin{pmatrix}-1 & 3 & 4 & 1 & -5 & 2\end{pmatrix}$.

Finally it should be mentioned that during this project basic approaches to prototypes for the following have been constructed.[95]

- A web interface for Markov generation of note sequences. The user inputs a chord structure into a user interface and one can easily generate some music using the Markov model. This is quite fast but way less advanced, because it doesn't include the optimization procedure. Sheet music can be generated within the browser by Verovio.[96]
- A "reason tracker" during the optimization. Every accepted change is tracked and the objectives that improved are connoted with a reason why the change has been accepted. For the Markov objectives one can easily find models in the corpus, for the correlation objective one can present models as well and for the Gradient Tree Boosting one can find the tree that improved most and then find the most important feature within that tree. Although it is not always the reason in a strict sense, at least it is some heuristic that could enable one to use the modification for some pedagogical setup.
- It has been tried to use the *Automated Analytics* view of *SAP Predictive Analytics*[97] to replace the Gradient Tree Boosting. It performs some kind of proprietary logistic regression that attempts to find the best model given some

94 *Musical Style*; *Virtual Music*; *Musical Creativity*. Also see section 1.1 on pp. 1 et seqq.
95 See appendix E for the code.
96 `http://www.verovio.org`
97 `http://go.sap.com/product/analytics/predictive-analytics.html`

data in a black box fashion. Although it does not perform as well as Gradient Tree Boosting in terms of the accuracy, in some settings it performed a remarkable feature selection (i.e. using few features for classifying). See appendix E for a digital report of these experiments. It is not included in the printed version of this thesis because after this comparison, the project was further developed, so it isn't a fair comparison with the final Gradient Tree Boosting results anymore, but it has been one then.

a: from *Donna Lee*
b: from *Donna Lee*
c: from *Donna Lee*
d: from *Donna Lee*
e: from *Donna Lee*
f: from *Donna Lee*
g: from *Donna Lee*
h: from *Donna Lee*
i: from *Donna Lee*
j: from *(Used To Be A) Cha Cha*
k: from *Slang*
l: from *The Days of Wine and Roses*
m: from *The Days of Wine and Roses*

Figure 50: Examples of Pastorius all containing the interval succession $\begin{pmatrix} -1 & -1 & -1 & 4 & 3 \end{pmatrix}$ with a tolerance of 2 semitones.

5 Conclusion

5.1 Suggestions for further research

There are dozens of ways of improving aspects of this thesis or adding new interesting perspectives. Here I want to focus on some aspects, that seem especially promising for me.

- Speed

The current implementation of the modification procedure is a proof of concept and quite slow. This has several reasons. Firstly the project was developed while programming, so the goal wasn't clear when starting and so speed was no priority. Rewriting the software anew with a clear goal surely would result in code, that is more clean and efficient. A first attempt would also be to port the project from Python 2 (current state) to Python 3.

Secondly this project, especially the feature extraction, heavily relies on music21. music21 is designed for being a solution to all computational-musicological problems, so the majority of it isn't needed within this project. So writing a custom, more specialized music representation and dropping music21 as dependency would clearly be a benefit in respect to speed. On the other hand that would considerably increase the amount of code to be maintained in this project.

Another way for gaining speed, would be to use a more directed local search procedure. E.g., if one would not restrict the note events to be discrete, but continuous one could apply some quasi-Newton method.[98] Afterwards the note events could be quantized again. However, that would require the computing of neighbors as well as the evaluation of the objectives to be quite fast already, because that is needed many times by this type of algorithm.

According to Philip N. Johnson-Laird[99] "there are only three sorts of algorithm that could be creative"[100]. In his terminology probably the project presented in the present thesis would be a neo-Darwinian one, which is inefficient according to

[98] Like the BFGS-method, see Jorge Nocedal/Stephen J. Wright: *Numerical Optimization*, 2nd ed., New York 2006, pp. 136 et seqq. with the approximated gradient, see ibid., pp. 195 et seqq.

[99] Philip N. Johnson-Laird: *How Jazz Musicians Improvise*, in: *Music Perception* 19.3 (2002), pp. 415–442.

[100] Ibid., p. 420.

him. The suggestion of his research would be to add neo-Lamarckian elements if the algorithm should create fully creative results.

- Paying more regard to the metrical structure

Since an obvious drawback of the modification results is the insufficient regard of the metrical structure, one could try to overcome this by an additional metrical objective. One could analyze the corpus for often reoccurring metrical patterns and reward similar patterns during the optimization.

- Relation of Jaco Pastorius and Charlie Parker

There could be more research about the relation of Jaco Pastorius and Charlie Parker when one would transcribe larger amounts of music by Parker. Then the queries presented in section 4 could be more fruitful. Besides searching for quotes one could also search for and analyze sections with similar feature distributions, which would open a new application of this thesis.

- Music generation based on a theme

Currently an initial state of the music has to be complete for a modification process. It would be interesting to extend it the way, that the initial state is only a theme with chord progression of a couple of bars and from this a larger "improvisation" is developed. This could have considerable use cases for jazz musicians.

- Using the modification pedagogical

In section in section 4 I already shortly described a way how to produce "reasons" for the accepted modification steps. By enhancing this and properly putting this into layman's terms, one could built a pedagogical application with possible use cases in practical music education.

- Evaluation of modification procedure

The evaluation of the results of the modification is very subjective. A systematic survey among a representative group – people with some experience with jazz at least – could give empirical evidence concerning the performance of the modification procedure.

5.2 Summary

In this thesis firstly a novel approach of style classification was presented. It consists mainly of Gradient Tree Boosting of several features, many of which are customly developed, and of chord based Markov models. The classification results are quite satisfying. Secondly, this classification is then used as objective, among others, in a local search for the sake of style modification. The objectives try to reward similarity to Pastorius' style as well as his type of large scale structure. By that means a given piece of music can be transformed with the aim of making it closer to the style of Pastorius. In addition a short introductory overview about some aspects of style modification was given, some additional research findings have been presented and possibilities for further research have been suggested.

The results of the modification exhibit the desired blending of the style of Pastorius into an existing piece of music. Nevertheless, usually the result doesn't achieve the quality of a real improvisation or composition, mainly because of the lack of regard to the metrical structure and the lack of data. However, it is considerably better than purely random modifications and one often can feel the style of Jaco in the modified results. This is approved by a bass player who also emphasized the potential of this method as a tool for learning a musical style. So this approach can be regarded as quite promising.

5.3 Closing remarks

To round off here is an insightful quotation from a completely different area: a definition of "stylistic copy" from the perspective of an art historian concerning German sculptures of the 16th and 17th century.

> However, as stylistic copy should be considered an adoption of form, that wants to give the aura of the ancient, of something longly overcome, something approved by time to a new work and thereby goes so far that a utopian identity between present and past is induced [...][101]

[101] Claude Keisch: *Zu einigen Stilkopien in der deutschen Plastik des 16. und 17. Jahrhunderts*, in: *Forschungen und Berichte*, Bd. 15, Kunsthistorische und volkskundliche Beiträge 1973, pp. 71–78, p. 71, translation by the author. Original version:

> Als Stilkopie hingegen sollte man nur eine Formenübernahme bezeichnen, die dem neuen Werk die Aura des Alten, seit langem Überkommenen, durch die Zeit Bestätigten vermitteln möchte und dabei so weit geht, eine utopische Identität

With this in mind, I hope this thesis partly enables to induce such a utopian identity. The method presented has been tailored to the monophonic bass guitar music of Jaco Pastorius, but it could also be extended to other monophonic types of music and different style, as appropriate with custom feature extractors and music data.

As for myself, I learned much during this project: Dealing with data science and symbolic music processing as well as working in an professional environment are valuable experiences that influenced me in a positive manner, not least qualified myself for a position as research fellow as from October at Saarland University where my activity with music processing will continue.

zwischen dem Gegenwärtigen und dem Vergangenen herzustellen [...]

Appendices

A Short titles

Artificial Intelligence	STUART RUSSELL/PETER NORVIG: *Artificial Intelligence. A Modern Approach*, 3rd ed., Harlow 2014.
Greedy Approximation	JEROME H. FRIEDMAN: *Greedy Function Approximation: A Gradient Boosting Machine*, in: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232.
Jaco Portrait	SEAN MALONE: *A Portrait of Jaco. The Solo Collection*, Milwaukee 2002.
Jazz Theory and Practice	RICHARD LAWN/JEFFREY HELLMER: *Jazz Theory and Practice*, Los Angeles 1996.
jMIR	CORY MCKAY: *Automatic Music Classification with jMIR*, online (accessed 4-August-2015), PhD thesis, Montreal, 2010, URL: `http://jmir.sourceforge.net/publications/PhD_Dissertation_2010.pdf`.
Markov Constraints	FRANÇOIS PACHET/PIERRE ROY: *Markov constraints: steerable generation of Markov sequences*, in: *Constraints* 16.2 (2011), pp. 148–172.
Metaheuristics	SEAN LUKE: *Essentials of Metaheuristics*, 2nd ed., Raleigh 2013, available at `http://cs.gmu.edu/~sean/book/metaheuristics`.
Musical Creativity	DAVID COPE: *Computer Models of Musical Creativity*, Cambridge 2005.
Musical Style	DAVID COPE: *Computers and Musical Style*, Oxford 1991.
Statistical Learning	TREVOR HASTIES/ROBERT TIBSHIRANI/JEROME H. FRIEDMAN: *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*, 2nd ed., New York 2009.
Virtual Music	DAVID COPE: *Virtual Music. Computer Synthesis of Musical Style*, Cambridge 2001.

B References

Abesser, Jakob: *Automatic Transcription of Bass Guitar Tracks applied for Music Genre Classification and Sound Synthesis*, PhD thesis, Technische Universität Ilmenau, 2014, url: `http://www.db-thueringen.de/servlets/DocumentServlet?id=24846`.

Adler, Guido: *Der Stil in der Musik. 1. Buch: Prinzipien und Arten des Musikalischen Stils*, Leipzig 1911.

Alpaydin, Ethem: *Introduction to Machine Learning*, 2nd ed., Cambridge and London 2010.

Biles, John A.: *GenJam: A Genetic Algorithm for Generating Jazz Solos*, in: *Proceedings of the 1994 International Computer Music Conference*, 1994, pp. 131–137.

Brooks, Frederick P., A. L. Hopkins, Peter G. Neumann, and William V. Wright: *An experiment in musical composition*, in: *IRE Transactions on Electronic Computers*, vol. 6, 1957, pp. 175–82.

Chen, Stanley F. and Joshua Goodman: *An Empirical Study of Smoothing Techniques for Language Modeling*, in: *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics* 1996, pp. 310–318.

Cope, David: *Computer Models of Musical Creativity*, Cambridge 2005.

Idem: *Computers and Musical Style*, Oxford 1991.

Idem: *Virtual Music. Computer Synthesis of Musical Style*, Cambridge 2001.

Cuthbert, Michael S., Christopher Ariza, and Lisa Friedland: *Feature Extraction and Machine Learning on Symbolic Music using the music21 Toolkit*, in: Anssi Klapuri and Colby Leider (eds.): *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011, pp. 387–392, url: `http://ismir2011.ismir.net/papers/PS3-6.pdf`.

Dannenberg, Roger B., Belinda Thom, and David Watson: *A Machine Learning Approach to Musical Style Recognition*, in: *Proceedings of the International Computer Music Conference*, 1997, pp. 344–347.

Dias, Rui and Carlos Guedes: *A Contour-based Jazz Walking Bass Generator*, in: *Proceedings of the Sound and Music Computing Conference* 2013, pp. 305–308.

Downie, J. Stephen: *The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research*, in: *Acoustical Science and Technology*, vol. 29, 2008, pp. 247–255.

Elith, Jane, John R. Leathwick, and Trevor Hastie: *A working guide to boosted regression trees*, in: *Journal of Animal Ecology* 77.4 (2008), pp. 802–813.

Eppler, Arndt, Andreas Männchen, Jakob Abesser, Christof Weiss, and Klaus Frieler: *Automatic Style Classification of Jazz Records with Respect to Rhythm, Tempo, and Tonality*, in: Timour Klouche and Eduardo Miranda (eds.): *Proceedings of the 8th Conference on Interdisciplinary Musicology (CIM14)*, 2014.

Fernández, Jose D. and Francisco Vico: *AI Methods in Algorithmic Composition: A Comprehensive Survey*, in: *Journal of Artificial Intelligence Research*, vol. 13, 2013, pp. 513–582.

Frandsen, Mark S.: *Forecasting fusion at low frequencies: The bass players of "Weather Report"*, PhD thesis, Texas Tech University, 2010, url: `http://repositories.tdl.org/ttu-ir/handle/2346/45478`.

Friedman, Jerome H.: *Greedy Function Approximation: A Gradient Boosting Machine*, in: *The Annals of Statistics* 29.5 (2001), pp. 1189–1232.

Frieler, Klaus, Jakob Abesser, Wolf-Georg Zaddach, and Martin Pfleiderer: *Introducing the Jazzomat Project and the Melo(S)py Library*, in: *Proceedings of the Third International Workshop on Folk Music Analysis* 2013, pp. 76–78.

Hasties, Trevor, Robert Tibshirani, and Jerome H. Friedman: *The Elements of Statistical Learning. Data Mining, Inference, and Prediction*, 2nd ed., New York 2009.

Hiller, Lejaren and Leonard Isaacson: *Experimental Music. Composition With an Electronic Computer*, New York 1959.

Janssen, Berit, W. Bas de Haas, Volk Anja, and Peter van Kranenburg: *Discovering repeated patterns in music: state of knowledge, challenges perspectives*, in: Mitsuko Aramaki, Olivier Derrien, Richard Kronland-Martinet, and Sølvi Ystad (eds.): *Sound and Muic. 10th International Symposium on Computer Music Multidisciplinary Research 2013*, 2014, pp. 277–297.

Jelinek, Frederick and Robert L. Mercer: *Interpolated Estimation of Markov Source Parameters from Sparse Data*, in: *Pattern Recognition in Practice. Proceedings of an International Workshop held in Amsterdam* 1980, pp. 381–397.

Johnson-Laird, Philip N.: *How Jazz Musicians Improvise*, in: *Music Perception* 19.3 (2002), pp. 415–442.

Jones, Merilyn: *An Investigation of Skills in Recognition of Musical Style Among College Freshman Music Majors*, in: *Contributions to Music Education* 16 (1989), pp. 77–86.

Keisch, Claude: *Zu einigen Stilkopien in der deutschen Plastik des 16. und 17. Jahrhunderts*, in: *Forschungen und Berichte*, Bd. 15, Kunsthistorische und volkskundliche Beiträge 1973, pp. 71–78.

Kotsifakos, Alexios, Evangelos E. Kotsifakos, Panagiotis Papapetrou, and Vassilis Athitsos: *Genre classification of symbolic music with SMBGT*, in: Fillia Makedon, Margrit Betke, Magy Seif El-Nasr, and Ilias Maglogiannis (eds.): *Proceedings of the 6th International Conference on Pervasive Technologies Related to Assistive Environments*, 2013, 44:1–44:7.

Kranenburg, Peter van: *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*, PhD thesis, Utrecht University, 2010, url: `http://dspace.library.uu.nl/handle/1874/179892`.

Lawn, Richard and Jeffrey Hellmer: *Jazz Theory and Practice*, Los Angeles 1996.

Luke, Sean: *Essentials of Metaheuristics*, 2nd ed., Raleigh 2013, available at `http://cs.gmu.edu/~sean/book/metaheuristics`.

Malone, Sean: *A Portrait of Jaco. The Solo Collection*, Milwaukee 2002.

McKay, Cory: *Automatic Music Classification with jMIR*, online (accessed 4-August-2015), PhD thesis, Montreal, 2010, url: `http://jmir.sourceforge.net/publications/PhD_Dissertation_2010.pdf`.

Meyer, Leonard B.: *Meaning in Music and Information Theory*, in: *The Journal of Aesthetics and Art Criticism* 15.4 (1957), pp. 412–424.

Milkowski, Bill: *Jaco: The Extraordinary and Tragic Life of Jaco Pastorius, "the World's Greatest Bass Player"*, Milwaukee 1995.

Neumann, Frederick: *Ornamentation in Baroque and Post-baroque Music. With special emphasis on J.S. Bach*, 3rd ed., Princeton 1983.

Nocedal, Jorge and Stephen J. Wright: *Numerical Optimization*, 2nd ed., New York 2006.

Orfanidis, Sophocles J.: *Introduction to Signal Processing*, online (accessed 2-July-2015), 2010, url: `http://www.ece.rutgers.edu/~orfanidi/intro2sp/`.

Pachet, François and Pierre Roy: *Markov constraints: steerable generation of Markov sequences*, in: *Constraints* 16.2 (2011), pp. 148–172.

Pachet, François, Pierre Roy, and Gabriele Barbieri: *Finite-length Markov Processes with Constraints*, in: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, 2011, pp. 635–642.

Papadopoulos, George and Geraint Wiggins: *A Genetic Algorithm for the Generation of Jazz Melodies*, in: *Proceedings of the Finnish Conference on Artificial Intelligence*, 1998, pp. 7–9.

Pfleiderer, Martin and Klaus Frieler: *The Jazzomat project. Issues and methods for the automatic analysis of jazz improvisations*, in: *Concepts, Experiments, and Fieldwork: Studies in Systematic Musicology and Ethnomusicology* 2010, pp. 279–295.

Pinkerton, R. C.: *Information theory and melody*, in: *Scientific American* 194.2 (1956), pp. 77–86.

Ridgeway, Greg: *Generalized Boosted Models. A guide to the gbm package*, 2012, url: `https://github.com/harrysouthworth/gbm/blob/master/inst/doc/gbm.pdf`, gradually developed online paper for the R gbm package, version from May 23, 2012.

Rosen, Charles: *The Classical Style: Haydn, Mozart, Beethoven*, New York 1995.

Rosen, Steve: *Portrait of Jaco*, in: *International Musician and Recording World* August 1978, available online via `http://jacopastorius.com/features/interviews/portrait-of-jaco/`.

Russell, Stuart and Peter Norvig: *Artificial Intelligence. A Modern Approach*, 3rd ed., Harlow 2014.

Schapir, Robert E.: *The Strength of Weak Learnability*, in: *Machine Learning* 5.2 (1990), pp. 197–227.

Shellard, Martin: *Charlie Parker for Bass. 20 Heads & Sax Solos Arranged for Electric Bass with Tab*, Milwaukee 2014.

Wooten, Victor: *The Best of Victor Wooten. Transcribed by Victor Wooten*, Milwaukee 2003.

Xenakis, Iannis: *Formalized Music. Thought and Mathematics in Composition*, ed. by Sharon Kanach, revised edition, Stuyvesant 1992.

Youngblood, Joseph E.: *Style as Information*, in: *Journal of Music Theory*, vol. 2, 1958, pp. 24–35.

C List of figures

D E-Mail from Hal Leonard Product Support

Sender	Hal Leonard Product Support <halinfo3@halleonard.com>
Sent	July 23, 2015 13:17 -0500 (CST)

Thank you for your email,

We spoke to our production editor and they were able to provide the following information for you:

Martin Shellard did the arrangements and can be considered the ‘author’. Martin worked from existing transcriptions from our various Charlie Parker publications.

We hope this helps with your research.

Sincerely,
Hal Leonard Product Support

Sender	Frank Zalkow <frank_zalkow@web.de>
Sent	July 23, 2015 11:09 -0500 (CST)

Dear Sir or Madam,

for my Master's thesis I am consulting "Charlie Parker for Bass" (Hall Leonard, 2014). For quoting I would like to know, who 1.) transcribed the songs and 2.) who arranged them for bass. I also would like to know who is considered the author or editor of this volume.

Thanks a lot for any information and best wishes,

Frank Zalkow

E Digital appendix

E.1 CD-R

E.2 Directory structure

```
/
├── modification ........... Contains MusicXML and MIDI files for the examples of section 3.4.
├── sklearn_vs_sappa ....... Contains files related to the comparison of SAP Predictive Analytics and scikit-learn's Gradient Tree Boosting, mentioned in section 4. See the file README.md there for further information, usage instructions and dependencies.
├── src .................... Contains the main source code of the thesis, including multiple scripts and utilities for different tasks. See the file README.md there for further information, usage instructions and dependencies.
│   ├── data_guitarpro ...... Contains the Guitar Pro data corpus, see appendix E.3, including several computed files from that, e.g. extracted features and files for the Markov model.
│   ├── data_transcribed .... Contains the transcribed data corpus, see appendix E.4, including several computed files from that, e.g. extracted features and files for the Markov model.
│   └── jacolib ............. Contains the main library, used all over the place in all scripts of this digital appendix.
├── transcriptions ......... Contains the Lilypond files that made up the transcribed data corpus as well as scripts for converting them into formats to be processed by other scripts. See the file README.md there for further information, usage instructions and dependencies.
│   ├── brahms
│   ├── escaida
│   ├── jaco
│   ├── parker
│   └── wooten
└── web_ui ................. Contains files for the browser interface for generating music by the Markov models, mentioned in section 4. See the file README.md there for further information, usage instructions and dependencies.
```

E.3 Pieces collected as Guitar Pro files

Marcus Miller

- Marcus Miller: *Bruce Lee*
- Marcus Miller: *Bruce Lee*
- Marcus Miller: *Frankenstein*
- Luther Vandross: *If Only for One Night*
- Marcus Miller: *Introduction*
- Boz Scaggs: *Lowdown*
- Marcus Miller: *Mr Pastorius*
- Luther Vandross: *Never Too Much*
- Marcus Miller: *People Make The World Go Round*
- Marcus Miller: *Power*
- Marcus Miller: *Rampage*
- Marcus Miller: *Run for Cover*
- Luther Vandross: *She's A Super Lady*

Jaco Pastorius

- Weather Report: *A Remark You Made*
- Jaco Pastorius: *Amerika*
- Weather Report: *Barbary Coast*
- Weather Report: *Birdland*
- Weather Report: *Black Market*
- Jaco Pastorius: *Blackbird*
- Pat Metheny: *Bright Size Life*
- Johann S. Bach: *Chromatic Fantasy*
- Jaco Pastorius: *Come On, Come Over*
- Jaco Pastorius: *Continuum*
- Charlie Parker *Donna Lee*
- Joni Mitchell: *Dry Cleaner From Des Moines*
- Weather Report: *Elegant People*
- Jaco Pastorius: *Havona*
- Jaco Pastorius: *Jam In E*
- Mike Stern: *Mood Swings*
- Jaco Pastorius: *Opus Pocus*
- Weather Report: *Palladium*
- Wayne Shorter: *Port Of Entry*
- Jaco Pastorius: *Portait Of Tracy*
- Jaco Pastorius: *Punk Jazz*
- Jaco Pastorius: *Reza*
- Jaco Pastorius: *Slang*
- Weather Report: *Teentown*
- Jaco Pastorius: *The Chiken*
- Jaco Pastorius: *(Used To Be A) Cha cha*
- Jaco Pastorius: *Word Of Mouth*

E.4 Transcribed pieces

Jaco Pastorius

- Pat Metheny: *Bright Size Life*
- Charlie Parker: *Donna Lee*
- Jaco Pastorius: *Havona*
- Wayne Shorter: *Port Of Entry*
- Jaco Pastorius: *Punk Jazz*
- Jaco Pastorius: *Slang*
- Henry Mancini: *The Days of Wine and Roses*
- Jaco Pastorius: *(Used To Be A) Cha Cha*

Charlie Parker

- Charlie Parker and Dizzy Gillespie: *Anthropology*
- Charlie Parker: *Moose the Mooche*
- Charlie Parker and Bennie Harris: *Ornithology*
- Charlie Parker: *Yardbird Suite*

Victor Wooten

- Victor Wooten: *A Show of Hands*
- Bela Fleck, Victor Wooten and Howard Levy: *Blu-Bop*
- Ray Noble: *Cherokee (Indian Love Song)*
- Victor Wooten: *Classical Thumb*
- John Lennon and Paul McCartney: *Norwegian Wood (This Bird Has Flown)*
- Victor Wooten: *Sex in a Pan*
- Bela Fleck: *Sinister Minister*
- Victor Wooten and Bela Fleck: *Stomping Grounds*

F Used feature extractors from music21

See Cory McKay's PhD[105] and music21's reference[106] for a description of the feature extractors. There is one dimension per feature if not stated otherwise.

`music21.features.jSymbolic.AmountOfArpeggiationFeature`
`music21.features.jSymbolic.AverageMelodicIntervalFeature`
`music21.features.jSymbolic.AverageNoteDurationFeature`
`music21.features.jSymbolic.AverageTimeBetweenAttacksFeature`
`music21.features.jSymbolic.AverageVariabilityOfTimeBetweenAttacksForEachVoiceFeature`
`music21.features.jSymbolic.BasicPitchHistogramFeature` (128 dimensions)
`music21.features.jSymbolic.ChangesOfMeterFeature`
`music21.features.jSymbolic.ChromaticMotionFeature`
`music21.features.jSymbolic.CompoundOrSimpleMeterFeature`
`music21.features.jSymbolic.DirectionOfMotionFeature`
`music21.features.jSymbolic.DistanceBetweenMostCommonMelodicIntervalsFeature`
`music21.features.jSymbolic.DurationFeature`
`music21.features.jSymbolic.DurationOfMelodicArcsFeature`
`music21.features.jSymbolic.FifthsPitchHistogramFeature` (12 dimensions)
`music21.features.jSymbolic.ImportanceOfBassRegisterFeature`
`music21.features.jSymbolic.ImportanceOfHighRegisterFeature`
`music21.features.jSymbolic.ImportanceOfMiddleRegisterFeature`
`music21.features.jSymbolic.InitialTimeSignatureFeature` (2 dimensions)
`music21.features.jSymbolic.IntervalBetweenStrongestPitchClassesFeature`
`music21.features.jSymbolic.IntervalBetweenStrongestPitchesFeature`
`music21.features.jSymbolic.MaximumNoteDurationFeature`
`music21.features.jSymbolic.MelodicIntervalHistogramFeature` (128 dimensions)
`music21.features.jSymbolic.MinimumNoteDurationFeature`
`music21.features.jSymbolic.MostCommonMelodicIntervalFeature`
`music21.features.jSymbolic.MostCommonMelodicIntervalPrevalenceFeature`
`music21.features.jSymbolic.MostCommonPitchClassFeature`
`music21.features.jSymbolic.MostCommonPitchClassPrevalenceFeature`
`music21.features.jSymbolic.MostCommonPitchFeature`
`music21.features.jSymbolic.MostCommonPitchPrevalenceFeature`
`music21.features.jSymbolic.NoteDensityFeature`
`music21.features.jSymbolic.NumberOfCommonMelodicIntervalsFeature`
`music21.features.jSymbolic.NumberOfCommonPitchesFeature`
`music21.features.jSymbolic.PitchClassDistributionFeature` (12 dimensions)
`music21.features.jSymbolic.PitchClassVarietyFeature`
`music21.features.jSymbolic.PitchVarietyFeature`
`music21.features.jSymbolic.PrimaryRegisterFeature`
`music21.features.jSymbolic.QuintupleMeterFeature`
`music21.features.jSymbolic.RangeFeature`
`music21.features.jSymbolic.RelativeStrengthOfMostCommonIntervalsFeature`
`music21.features.jSymbolic.RelativeStrengthOfTopPitchClassesFeature`
`music21.features.jSymbolic.RelativeStrengthOfTopPitchesFeature`
`music21.features.jSymbolic.RepeatedNotesFeature`
`music21.features.jSymbolic.SizeOfMelodicArcsFeature`
`music21.features.jSymbolic.StaccatoIncidenceFeature`
`music21.features.jSymbolic.StepwiseMotionFeature`
`music21.features.jSymbolic.TripleMeterFeature`
`music21.features.jSymbolic.VariabilityOfTimeBetweenAttacksFeature`
`music21.features.native.FirstBeatAttackPrevalence`
`music21.features.native.MostCommonNoteQuarterLengthPrevalence`
`music21.features.native.MostCommonNoteQuarterLength`
`music21.features.native.RangeOfNoteQuarterLengths`
`music21.features.native.TonalCertainty`
`music21.features.native.UniqueNoteQuarterLengths`

[105] *jMIR.*
[106] `http://web.mit.edu/music21/doc/moduleReference/`