Subdivision-Based Parsing of Expressively Performed Rhythms

Bastiaan van der Weij

Master of Science
Cognitive Science and Natural Language Processing
School of Informatics
University of Edinburgh
2012

Abstract

An approach is presented here that interprets rhythmic structure of monophonic music performances based on onset times. Rhythmic structure is represented as subdivision trees. A chart parsing algorithm that uses probabilistic beam search is presented to construct these structures. A Bayesian probabilistic model is used in which the probability of a performance given a rhythmic structure is modelled by an expression model and the probability of the rhythmic structure itself is modelled by a rhythm model. The system is completely tempo-independent and therefore well-suited for studying expression.

It is suggested that local expressive deviations, such as stretching certain beats to emphasise them, are related to rhythmic structure. An expression-aware model is proposed that is sensitive to these deviations and it is suggested that an expression-aware model may improve parser performance. In addition to that, an alternative expression model is proposed that treats expressive deviation as noise.

Furthermore, a new corpus containing monophonic performances of well-known jazz standards annotated with rhythmic structure is presented here and used to evaluate the parser.

The performance of both our expression models is compared to a baseline. This baseline uses an uninformed rhythm model that ranks every rhythm as equally likely and an expression model that treats expressive deviation as noise. It was found that while our expression-aware model did not perform better than the baseline, the alternative expression model did.

Acknowledgements

I am thankful to Mark Steedman and Mark Granroth-Wilding for their guidance, support and ideas. Without this and without the trust they put into my completion of the work, I would have been unable to produce this thesis.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Bastiaan van der Weij*)

Table of Contents

Chapter 1

Introduction

The work in this thesis is concerned with the analysis of the rhythmic structure in performances of music. We will study rhythmic structure in isolation of other structures that may be present in the music like melody or harmony.

This chapter will introduce concepts used throughout this thesis and discuss related studies. First, rhythmic structure is introduced. Then in section 1.2, structure will be related to performance and finally, in section 1.3, the approach presented in this thesis will be introduced.

The rest of this thesis is structured as follows: In chapter 2, our approach will be described in detail. Chapter 2 will also introduce the annotated jazz corpus that was produced for this thesis. In chapter 3, we will describe how we intend to evaluate our system. Then in chapter 4 the system will be evaluated on the jazz corpus. Chapter 5 will discuss to what extent the system was successful based on the results and improvements will be suggested. Finally, chapter 6 will present the conclusions of this thesis.

1.1 Rhythmic Structure

In most Western music traditions, a rhythm is constructed of notes and rests with some metrical duration. A note tells the performer to play some pitch for the duration of the note, while a rest indicates a silence for the duration of the rest. A metrical duration can be subdivided into a prime number of beats. The first of these beats is called the *downbeat*, the rest are called *upbeats*.

In the so-called staff notation, metrical units have been given names and durations are specified as subdivisions of the duration of what is called a whole note: half notes

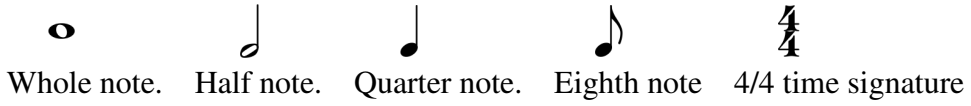


Figure 1.1: Some music notation conventions.

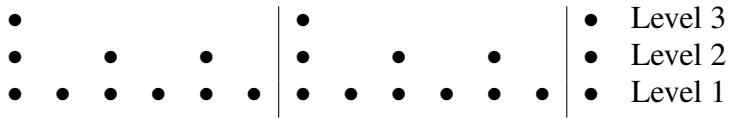


Figure 1.2: Metrical grid of a 3/4 time signature.

or rests, quarter notes or rests, eighth notes or rests, etcetera. Their notation is shown in figure 1.1. Apart from these duple divisions, triple, quintuple and any other prime number divisions are allowed as well. These divisions are called tuplets. Duple and triple divisions are by far the most common in Western music.

Metrical durations are grouped together in units called bars or measures. A *time signature* specifies how a bar is subdivided into metrical units. The time signature consists of two numbers written above each other. A 4/4 time signature is shown on the far right in figure 1.1. The top number specifies the number of beats per measure, the bottom number specifies the units of those beats. The time signature is sometimes called meter.

Staff music notation is one of the most widespread representations of rhythmic structure. In the field of rhythm analysis, another structure is popular, called a *metrical grid*, originally introduced by Lerdahl and Jackendoff (1983) in their Generative Theory of Tonal Music. A metrical grid is a representation that contains several levels. The lowest level corresponds to the smallest subdivision and the highest level corresponds to a bar. Going down one level corresponds to a subdivision. A metrical grid representing a 3/4 time signature is shown in figure 1.2. Bars are separated using a vertical line, • symbols indicate the downbeats of each level. Level three, for example, contains one downbeat per bar. Level three is subdivided into three level-two downbeats, the second and third of which is a level-three upbeat. Every level-two unit is subdivided into two level-one downbeats, the second of which is a level-two upbeat. The time signature specifies that a bar is divided into three quarter notes, so level two corresponds to quarter notes in this case.

Having now introduced two representations of rhythmic structure, we can turn to some of the work done in this field. The most extensive recent study was conducted by Temperley (2010), who studied the probabilistic properties of rhythmic structures

in isolation from melody and harmony. It is suggested that rhythm has some probabilistic characteristics that are shared to some extent by rhythms in a wide range of musical styles. Temperley identifies a number of intuitions about rhythm that seem to be common practice. These intuitions include the general tendency of onsets to fall on downbeats, the preference for onsets on upbeats to be preceded or followed by a note on the previous or next downbeat and the tendency of long notes to fall on downbeats.

Temperley compares six different models intended to be sensitive to these regularities. The adequacy of these models is evaluated by measuring their cross-entropy, using cross-validation on a corpus of European folk songs. Temperley's work shows that probabilistic models can successfully differentiate sensible rhythms from nonsensical rhythms. Intuitively, these models explain that not every pattern of onsets that can be described as a valid rhythmic structure will be perceived as a rhythm by humans.

1.2 Performances

Rhythmic structure in itself does not define the absolute timing of notes. The common way to relate a rhythmic structure to a performance is by assigning some real duration to a metrical unit. The amount of real time we assign to a metrical duration is usually called the *tempo*. Given a tempo, a rhythmic structure can be converted to a set of *idealised* onset times, also called *metronomic* onset times. Onset times are the time at which a performer started playing a note, measured from the beginning of the performance.

When humans perform a rhythm, they deviate from the idealised onset times in several ways. Unless a metronome is used, the tempo will usually fluctuate as the performance progresses. Much of this fluctuation is intentional and is referred to as *musical expression*. Apart from global tempo changes, humans deviate from idealised onsets locally as well, even when a metronome is used.

In general, it is thought that, depending on the competence of the performer, some proportion of this local deviation, is noise but a large proportion of it seems to be systematic. A study by Palmer (1989) suggested that global tempo changes are mostly guided by conscious intention. Local deviations in timing and loudness seemed to be partly unconscious and represented in the performers mind in abstracted form. Pianists were for example aware that they articulated certain beats but could not reliably tell whether they did so by playing them louder or by altering their timing. These findings suggest that it is to some extent not even possible for a human to perform rhythm

without expression.

It seems that while global tempo changes may be guided by conscious intentions of phrasing, local expression may be partially unconscious and unavoidable. Another study has shown that there is some regularity in local expression that is linked to rhythmic structure (Bengtsson and Gabrielsson, 1983). For example in Vietnamese waltzes, which have almost exclusively a 3/4 or 6/8 meter, it is observed that there is a consistent lengthening of the first upbeat at quarter note level.

Although a performance deviates from the idealised onsets given by the structure and tempo, human listeners are often able to perceive the rhythmic structure in a performance and multiple listeners tend to be quite consistent in their structural interpretations of a performance. In fact, the studies mentioned earlier suggest that deviations from idealised onsets are crucial to perception of structure in rhythm.

Several authors have suggested models that try to mimic this human capacity. Cemgil et al. (2000) propose a system for rhythm quantisation that uses Bayesian modelling to derive the rhythmic structure. Their model tries to optimise the probability of a score given a performance, which can be expressed as the probability of the score (the score model) times the probability of the score given the performance (the rhythm model). The model assumes tempo can change and uses a tempo-track (often called tempo-curve) to derive metronomic onset times given local tempo. Their performance model simply penalises performances to the extent that they deviate from metronomic onset times.

Raphael (2002) proposes another model that is similar to the model of Cemgil et al. (2000). Here a graphical probabilistic model is proposed where 'score-positions' (relative positions of notes within a measure) are seen as a Markov chain of events where the probability of a score-position depends on the probability of a note on that score-position and the probability of a note on that score-position preceded by the score-position of the previous note. Another Markov chain of tempo values is generated from the the score positions and finally the exact onset time of each note is dependent on the tempo value, score position of the previous note and score position of the current note.

These models are discussed by Temperley (2007) and it is observed their representation of rhythm is still ambiguous with respect to meter. Temperley proposes a model of rhythm that is based on a metrical grid, an unambiguous representation of meter and also contains fewer parameters than the rather complicated models of Raphael (2002) and Cemgil et al. (2000). This model is elaborated in Temperley (2009) where

it is integrated into a unified probabilistic model of rhythm, harmony and polyphonic structure.

In the latter study, a rhythm model is implemented based on a concept that Temperley calls metrical anchoring: the probability of a note on an upbeat depends on whether a note was played on the preceding or following downbeat. Temperley calls this model the hierarchical position model. Later, in Temperley (2010), this model is compared to other rhythm models and it is shown that it performs well in comparison.

The hierarchical model uses metrical grids as a representation of rhythm. These grids are limited to four levels and contain only duple divisions. To deal with the problem of tempo, the model has a parameter that specifies the preferred *tactus level* interval. The tactus level corresponds to the tactus, or pulse, in a performance which often corresponds to quarter notes. The tactus interval is allowed to fluctuate: the probability of a next tactus interval is a normal distribution over the previous one, preferring tactus intervals of the same length.

The notion that performances of rhythm always seem to be to some extent expressive and that this expression may help humans perceiving their structure has been largely ignored by the models described above. In all of these models, expression was considered additive noise with respect to the idealised onsets given some rhythmic structure and tempo representation.

Another potential criticism of the models above is that they all include one or more parameters related to tempo. Tempo curves introduce extra parameters in the model. And since tempo fluctuates constantly in human performances, they never seem to be accurate enough. Tempo always has to be averaged over a number of notes, we can for example average the tempo per measure and consider any deviations from the tempo by individual beats to be expressive deviations. However, in a way, these deviations can be seen as changes in local tempo as well. It is not possible to estimate local tempo for every onset. This leads us to believe that the concept of tempo may not be appropriate in relation to expressively performed rhythms. A related view is put forward by Desain and Honing (1993), who argue that tempo curves can be a misleading concept.

In the next section, we will introduce a model that is completely tempo independent and has the potential to become sensitive to rhythmic structure-dependent local expression.

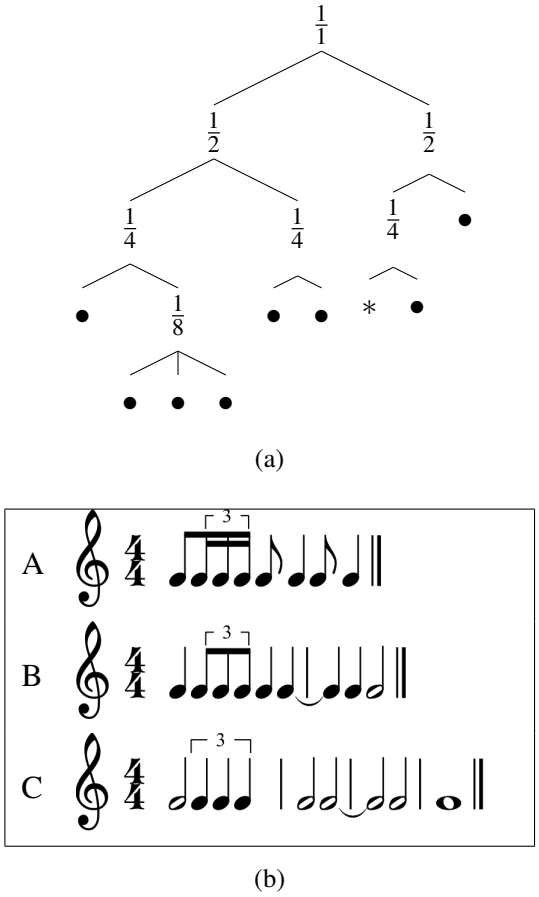
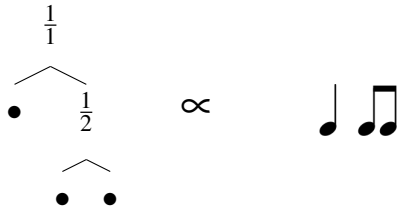


Figure 1.3: An analysis and three different score notations of the same musical cliché.

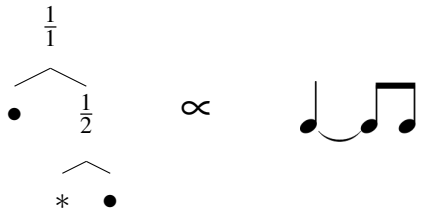
1.3 A Subdivision-Based Rhythmic Parser

The approach introduced here is inspired by the work of Longuet-Higgins (1976). In this work, rhythmic structure is represented as subdivision trees rather than staff notation or metrical grids. A subdivision tree represents rhythms as a hierarchical structure where some metrical duration corresponding to the root node is subdivided into child nodes. Figure 1.3a shows an example of such a structure and figure 1.3b shows three possible staff notation interpretations of the structure. Although musicians might play score C slower than score B and score B slower than score A, they are structurally identical and as we have said earlier, our model will be tempo-independent. The different scores are inferred from the tree by defining some level in the tree to correspond to the bar duration.

In modern staff notation the longest metrical unit is a whole note, which can be lengthened using dotted notation or ties. In the subdivision tree in figure 1.3, every node represents some metrical duration which may correspond to notes, bars or mul-



(a) A subdivision of a three note rhythm.



(b) A subdivision tree of a two note rhythm containing a tied note.

Figure 1.4: The interpretation of onsets and ties in subdivision trees.

tiple bars. Although figure 1.3 only shows duple and triple subdivisions, beats can be subdivided into any prime number of child-units. We specify two types of leaf-nodes: an onset, which we show as a $\bullet$ symbol in subdivision trees and a tied note, which we show as a $*$ symbol in subdivision trees. Figure 1.4 shows how these symbols are interpreted. The $\propto$ symbol means that the expression on the left is equivalent to the expression on the right except for any global scaling of note durations.

Longuet-Higgins (1976) proposed a rule-based system that needs to be initialised with a tactus interval and some tolerance parameter which specifies the rate at which the beat duration, and indirectly tempo, is allowed to fluctuate. Nowadays, computers are powerful enough to construct probabilistic models based on corpora and to consider a great number of possible structural interpretations of a performance at once. With this in mind, we will briefly introduce our system below before discussing it in detail in the next chapter.

We will boil down a performance of a rhythm to a list of onsets. We think onsets provide enough information to correctly identify rhythmic structure. Therefore, we will from now on talk about onsets rather than notes.

Subdivision trees representing valid rhythmic structures can be described in a context-free grammar (CFG). An algorithm that determines the hierarchical structure of an input given some CFG is called a parser. We will present a CFG and a parser that constructs hierarchical structures like in figure 1.3a from performances. Our parser will

be some variant of a class of efficient parsing algorithms called chart-parsers.

Our parser will be guided by a Bayesian model which allows us to define the probability of a rhythm given a performance as the probability of the rhythm itself, described by a *rhythm model* times the probability that this rhythm generated the observed performance, described by an *expression model*.

Similar to Temperley (2009), we will use a rhythm model that is sensitive to common-practice notions about rhythm. However, instead of his hierarchical model, which is based on metrical grids, we will use a probabilistic context-free grammar (see section 2.3). This model follows naturally from our representation of rhythmic structure. In chapter 5, we will evaluate how the PCFG prior compares to the hierarchical model.

We have claimed that some expressive deviations may have some structure-related regularities. Emphasising beats can result in a slight stretching of the duration of the beat. Depending on the style of the music, some beats may be emphasised consistently. It is often said that downbeats are emphasised in many music styles. If this does indeed lead to stretching them, it will result in downbeats at levels near the leaf nodes of the tree to be slightly longer than upbeats. A tempo independent way to look at this is as the ratio of downbeat length and upbeat length. Our expression model will assume that the down-/upbeat length ratio can be estimated from a small feature set. We will also present an alternative expression model that simply expects down-/upbeat ratios to be one-to-one and any deviation from this ratio is treated as noise.

To train our rhythm and expression model, we need a corpus of monophonic performances annotated with subdivision trees. There are, as far as we know, very few corpora available where expressively performed music is annotated with metrical structure and there are no corpora where this structure is represented as subdivision trees. There is a publicly available corpus of classical piano music performances of very high quality, annotated with tempo and deviation information for every performed note (Hashida et al., 2008). There are similar corpora in existence, notably one relatively large corpus containing annotated performances of Chopin's works by Nikita Magaloff (Flossmann et al., 2010), but these are not publicly available due to issues with copyright.

So although there is one freely available corpus, this corpus covers classical music. We think that we are more likely to find some form of regular expressive deviation in music that is generally performed at a relatively constant tempo. Jazz music seems to fit this requirement. As far as we know, there are no freely available corpora of jazz music annotated with rhythmic structure and therefore we endeavoured to create our

own corpus.

The corpus we constructed contains monophonic jazz melodies, performed by amateur musicians. The corpus was annotated with metrical onset times and subdivision trees were generated using a non-probabilistic version of our parser. This was possible because we could assume the onset times were metronomic.[1]

We will evaluate the parser on the jazz corpus. Since we do not have enough data to keep a separate test set, we have to use cross-validation. An implication is that we are effectively testing on our development set.

[1] Technically, the onsets are specified in metrical units and not metronomic units. However, since our parser is tempo-independent this does not matter.

Chapter 2

Method

In this chapter, the complete framework will be described. Section 2.1 will introduce a parser that constructs valid rhythmic structures. Section 2.2 will describe the probabilistic elements of the parser. These are elaborated in section 2.3 and 2.4 where respectively the rhythm and expression model are discussed. Section 2.5 will introduce the jazz corpus. Section 2.6 will describe how the rhythm and expression model are trained on the corpus. Finally, section 2.7 will describe a few implementation details of the parser.

2.1 Parsing Rhythms

We represent the performance P of a rhythm as series of note onset times.

$$P = [\mathrm{On}_0, \mathrm{On}_1, \cdots, \mathrm{On}_n] \tag{2.1}$$

A subdivision tree R is a hierarchical representation of rhythmic structure.

The approach presented here generates the most likely rhythmic structure R underlying a performance P. During this process the parser considers all possible hypotheses of sub spans of P and retains the most likely hypotheses, while rejecting the unlikely ones. We think this is computationally feasible because we assume that over a few notes, only a small number of hypotheses are worth considering. In this section we will first describe how structurally sound rhythmic analyses are generated. After that, we will outline a Bayesian model that defines how we determine whether a hypothesis is likely.

The parser we use is a slightly modified stochastic CKY chart parser (Younger, 1967). The full algorithm and modifications are given in appendix A. A small context-

free grammar augmented with some constraints will be used to generate subdivision trees. The grammar below constructs subdivision trees from onsets ($\bullet$) and ties ($*$).

$$\begin{aligned} R &\rightarrow R\,R \\ R &\rightarrow R\,R\,R \\ R &\rightarrow \bullet \\ R &\rightarrow * \end{aligned} \tag{2.2}$$

Every rhythmic structure, denoted by the R symbol corresponds to some metrical duration. A rule expansion for some R in the grammar above corresponds to subdividing the metrical duration of R into the number of symbols the rule expands into. For this study, we will restrict ourselves to duple and triple subdivisions.

The CKY parser only accepts grammars that are given in the so-called Chomsky normal form (CNF). That is, all rules should be of the form $A \rightarrow B, C$ or $A \rightarrow \alpha$, where A, B and C are non-terminal symbols and α is a terminal symbol. Converting the grammar above to CNF results in the following grammar:

$$\begin{aligned} R &\rightarrow R\,R \\ R &\rightarrow R\,R' \\ R' &\rightarrow R\,R \\ R &\rightarrow \bullet \\ R &\rightarrow * \end{aligned} \tag{2.3}$$

Two constraints are necessary to prevent the parser from generating invalid rhythmical structures. These constraints are: (1) Any set of two or three metrical durations are not allowed to combine if the first one expands directly to an onset and the others do not recursively expand to an onset; (2) Metrical durations are not allowed to combine if none of them recursively contains an onset.

The first constraint prevents the parser from generating structures with an onset on the downbeat and a tie on the upbeat. An upbeat tied to a downbeat is redundant: If we tie an upbeat quarter note to a downbeat quarter note we get a half note and not two tied quarter notes. The second constraint prevents the parser from combining subdivision trees that do not contain onsets. Figure 2.1a illustrates the first constraint and figure 2.1b illustrates the second.

Because we do not know how many $*$ symbols are present in the input, we had to modify the parser to also consider inputs with $*$ symbols added in places where it would satisfy the two constraints above. See appendix A for details.

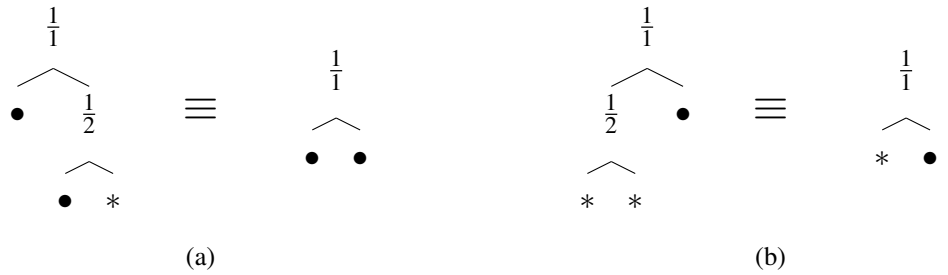


Figure 2.1: Redundant tree structures.

At this point, the parser generates all possible interpretations of an onset list of a certain length, which is an infinite amount. The next sections will describe how a probabilistic model is used to reject unlikely interpretations and retain the likely ones.

2.2 Hypothesis Generation and Rejection

Subdivision trees generated by the parser can be seen as hypotheses about the rhythmic structure of (some sub-span of) the performance. In our Bayesian model, the likelihood of a hypotheses given a set of performed onsets is determined by two factors: how likely is it that the rhythmic structure generated the observed performance and how likely is the rhythmic structure itself. In other words, we want to find the *posterior* probability $P(R|P)$, where P is a performance, and R is the rhythmic structure. We can formulate this as a generative model where

$$P(R|P) \propto P(P|R)P(R). \tag{2.4}$$

The posterior probability $P(R|P)$ of a rhythm R given a performance P is proportional to $P(P|R)$, the probability that R generated performance P times $P(R)$, the probability of R itself. $P(P|R)$ is called the *likelihood* of R given P and $P(R)$ is called the *prior* probability of R.

Another way to refer to the prior and the likelihood is respectively as a rhythm model and an expression model. The rhythm model should reflect intuitions about rhythms, for example that long notes tend to fall on downbeats, that duple divisions are more likely than triple divisions, etcetera. The prior used in this study will be described in section 2.3

The expression model defines how and to what extent we expect onsets to deviate from their expected onsets. In our system, the expression model will be based on one observation, which we shall call the *expression ratio*. The expression ratio is

defined as the logarithmic ratio of downbeat length and upbeat length. In metronomic performances, we expect downbeats and upbeats to be of equal length and their ratio to one. In human performances however, the expression ratio will be a measure of expressive timing. At low levels, the expression ratio reflects local expressive timing. A slightly stretched downbeat at the quarter note level for example, will produce an expression ratio slightly above zero. At higher levels the expression ratio reflects global changes in tempo. For example, slowing down gradually will result in expression ratios slightly below zero on higher levels.

Finally, the chart parser should only keep track of a limited number of sensible hypotheses. We will restrict hypothesis maintained by the parser in two ways: First, the per-item likelihood of the hypothesis (see section 2.6) should be higher than a certain threshold parameter. Second, after a cell has been filled with hypotheses by the parser (see appendix A for more details), hypotheses are ranked by their posterior probability and only the top-n hypotheses are kept. This both of these techniques implement a technique called 'beam search'.

Both the rhythm and the expression model will be trained on the corpus of annotated jazz performances that was constructed for this study and which will be described in detail in section 2.5.

2.3 The Rhythm Model

Our rhythm model will be a probabilistic context-free grammar (PCFG). A PCFG is a context-free grammar extended with probabilities for every rewrite rule. The probability of a syntax tree, produced by a PCFG can be derived by taking the product of every rule that was applied to construct the tree. In linguistics, PCFGs do not always assign probabilities to rules expanding to terminal symbols (words) since there are too many of them. In our case however, there are only two terminal symbols so we can assign probabilities to rules expanding to onsets or ties as well.

Note that there is only one non-terminal symbol in our grammar, namely R, so the probability of a rule expansion is given by:

$$P(R \rightarrow S) = P(S) = \frac{\text{count}(S)}{N}, \tag{2.5}$$

where S is a string of symbols and N is the total number of R symbols in the training set.

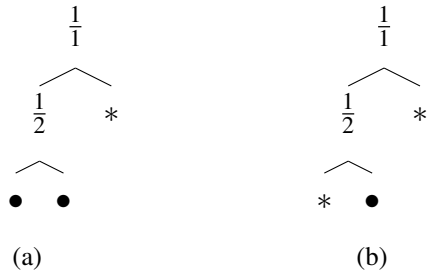


Figure 2.2: Two simple subdivision trees.

The probability of a subdivision tree R can be described as

$$P(R) = \prod_{R' \to S \in R} P(R' \to S). \tag{2.6}$$

where $R' \to S \in R$ refers to every rule expansion of some symbol R' to a string of symbols S that is recursively contained in R.

2.4 The Expression Model

This section will describe an expression model that represents expressive deviation as logarithmic ratios of downbeat and upbeat length. Section 2.6 will describe how this model is trained. Using different training methods, one model will be made 'expression-aware' and the other will treat expression as noise.

Before turning to how we observe expression ratios, we will describe how hypotheses are represented. Subsequently, we will introduce the features used to predict expression ratios and finally we will introduce the OBSERVATIONS function, which returns a set of features and expression ratios for any hypothesis.

In order to be able to say anything about the likelihood of an analysis, we need to have information about the onsets it contains. A subdivision tree represents rhythmic structure, but it does not keep track of the onsets associated with the structure. Therefore we will introduce a distinction between a subdivision tree or, rhythmic structure R, produced by the grammar in (2.3) and a *hypothesis* h.

A subdivision tree is a hierarchical structure that can be represented as a nested list. The tree in figure 2.2a for example, can also be written as $((\bullet,\bullet),*)$ and the tree in figure 2.2b as $((*,\bullet),*)$. Hypotheses have a similar structure but instead of onset symbols $\bullet$, contain actual onset times at the leaf nodes. We refer to the child nodes of an hypothesis as h_i where i is the beat position of the node so that h_0 is the

downbeat and h_i, where $i > 0$, are the upbeats. A hypothesis is said to *govern* an onset if it recursively contains that onset. The number of child nodes of a hypothesis corresponds to its subdivision. We will call the child nodes of a hypothesis its beats. Since subdivision trees can combine measures and groups of measures into metrical units, a beat can govern multiple measures.

We will define three functions over hypotheses: DIVISION(h), ONSETS(h) and BEATS(h). The DIVISION(h) function counts the number of child nodes, or beats, of hypothesis h, which can be two or three in our implementation. The ONSETS(h) function returns a list with length DIVISION(h) which contains the onset time or a tie symbol for each beat in the hypothesis. The list of onsets is constructed by taking the downbeat of every child node. The downbeat of a hypothesis can be defined recursively:

$$\text{DOWNBEAT}(h) = \begin{cases} \text{DOWNBEAT}(h_0) & \text{if } h \text{ has child nodes} \\ h & \text{otherwise} \end{cases} \tag{2.7}$$

Finally, BEATS(h) is a function that returns a list of expected onsets of every beat in h for all hypotheses that govern more than one onset. Predicting onset times is a bottom-up process which we will leave to appendix B to explain. If h governs a single onset, the BEATS and ONSETS functions are equivalent. For the tree in figure 2.2b for example, both the BEATS and the ONSETS function return $[*, *]$.

If h is represents an onset or tie, DIVISION(h) will return zero and BEATS(h) and ONSETS(h) will return h.

A top-down process will now determine the observed expression ratios and feature vectors that we mentioned earlier. This will be done by a function called OBSERVATIONS, which, given some hypothesis, returns a vector of observations containing feature vector/expression ratio pairs. The likelihood of these observations given their feature vectors can be determined after we learn to map feature vectors to an expected expression ratio. This will be done by training our model on the jazz corpus as described in section 2.6.

The feature vector will contain two features. We have already observed in section 2.2 that the expression ratio reflects different concepts as different levels, therefore one feature will be the `level` at which the expression ratio was observed. We will use another feature reflecting the `division` of the metrical unit in which the expression ratio was observed. The resulting feature vector is:

$$\varphi = [\texttt{level}, \texttt{division}]. \tag{2.8}$$

The `level` feature is defined bottom-up: level one is the deepest level of the tree. The `level` feature is not to be confused with depth: the highest level of a tree equals the root node, which has the shallowest depth.

For any hypothesis h governing more than one onset, we can calculate the expression ratio given that we know the (estimated or actual) onset of the downbeat, h_0 and the (estimated or actual) next downbeat. Since a hypothesis can be divided into more than two units, there may be more than one expression ratio defined per hypothesis. We define the *relative position* of a beat to be zero for the downbeat, one for the first upbeat and two for the second upbeat. The expression ratio for a hypothesis with division $d = \text{DIVISION}(h)$, onsets $O = \text{ONSETS}(h)$, predicted beats $B = \text{BEATS}(h)$, next downbeat B_d and upbeat onset O_i is calculated in the following fashion:

$$\text{expression ratio} = \log\left(\frac{(O_i - B_0)/i}{(B_d - O_i)/(d-i)}\right) \quad \text{for every } i \text{ where } i > 0 \text{ and } O_i \neq *. \tag{2.9}$$

The OBSERVATIONS function is used to derive downbeat and next downbeat estimates for any hypothesis that contains more than one onset and all the hypothesis recursively contained by h. This process has been designed with one intuition in mind: that downbeat intervals provide the most reliable information about where onsets are to be expected.

The OBSERVATIONS function is a top-down recursive function. It is initialised with a hypothesis h, the downbeat of h and the onset of next downbeat. Since we have not yet observed the next downbeat for the root-node hypothesis, we can only calculate the expected onset of the next downbeat. So for a hypothesis h, the algorithm is initialised as follows:

$$\text{OBSERVATIONS}(h, B_0, *, \text{DIVISION}(h) \times (B_1 - B_0)),$$

where $B = \text{BEATS}(h)$, the second argument B_0, is the downbeat and the last argument is the expected onset of the next downbeat.

The full OBSERVATIONS function is given in algorithm 1. The rest of this section will consist of a step-by-step description of this algorithm.

The algorithm starts with the initialisation of the subdivision of the hypothesis d, the list of beats B and list of onsets O. The estimated duration l of the hypothesis can now be calculated as the onset of the next downbeat minus the onset of the downbeat. Finally, the set of feature vector/expression ratio pairs S is initialised as an empty set and the $*$ symbol is appended to the list of onsets.

Now the algorithm iterates through every beat in the hypothesis. For every beat position i where $i > 0$ and $O_i \neq *$ the expression ratio is calculated as in equation 2.9. The requirements $i > 0$ and $O_i \neq *$ ensure that the expression ratio is only calculated for upbeats that contain actual onsets.

For every beat position in h, the algorithm will calculate the downbeat and next downbeat onset for the nested hypothesis at that position, these values are stored in b_{down} and b_{up}. For some beat position i, where $0 < i < d$, b_{down} is estimated by

$$b_{\text{down}} = \text{downbeat} + l * i/d. \tag{2.10}$$

If an onset has been estimated by the BEATS function, which indicated by $B_i \neq *$, this onset is used instead. Since the combination process will always have estimated onsets for hs governing more than one onset, equation 2.10 is only used when h governs a single onset.

Given b_{down} the position of the next downbeat, b_{up} is estimated by:

$$b_{\text{up}} = b_{\text{down}} + l/d. \tag{2.11}$$

This estimate should be equivalent to the onset at position $i+1$. If there is an onset at position $i+1$, this onset is preferred to the estimated onset.

Finally, the recursive step of the algorithm calls OBSERVATIONS$(h_i, b_{\text{down}}, b_{\text{up}})$ for every nested hypothesis h_i of h where $0 \leq i < d$ and DIVISION$(h) > 0$.

2.5 Data Preparation

To train the parser's rhythm model and expression model, a corpus of amateur jazz performances was prepared. The corpus contains jazz and latin standards that were scraped off the the web by Granroth-Wilding (forthcoming). The performances are generally of good quality and are played in a relatively constant tempo. In its original form, the corpus was a set of multi-track MIDI files containing both tracks played by a human performer and tracks generated by a computer in metronomic time.

MIDI files represent music as a list of note-on and note-off events, corresponding to key presses and key releases. Every event has the parameters pitch, on-velocity, off-velocity and delta-time. Delta-time specifies the time between the current event and the last one, pitch is the pitch of the key-press the current event corresponds, on-velocity is the velocity with which the key corresponding to the current event was pressed and off-velocity the velocity with which it was released.

Algorithm 1 Generate observations

function OBSERVATIONS(h, downbeat, nextDownbeat, expected)
  $B \leftarrow$ BEATS(h)
  $O \leftarrow$ ONSETS(h)
  $d \leftarrow$ DIVISION(h)
  $l \leftarrow$ nextDownbeat - downbeat
  $S \leftarrow \emptyset$
  append $*$ **to** O
  for $i \leftarrow 0, d$ **do**
    if $O_i \neq *$ **and** $i \neq 0$ **then**
      $\varphi \leftarrow$ (DEPTH(h), d)
      $r \leftarrow$ EXPRESSION_RATIO(downbeat, nextDownbeat, B_i, i, d)
      append (φ, r) **to** S
    end if
    $B' \leftarrow$ BEATS(h_i)
    $O' \leftarrow$ ONSETS(h_i)
    if DIVISION(h') $\neq 0$ **then**
      $b_{\text{down}} \leftarrow \text{downbeat} + l * i/d$
      if $B_i \neq *$ **then**
        $b_{\text{down}} \leftarrow B_i$
      end if
      $b_{\text{up}} \leftarrow b_{\text{down}} + l/d$
      if $O_{i+1} \neq *$ **then**
        $b_{\text{up}} \leftarrow O_{i+1}$
      end if
      append OBSERVATIONS($h_i, b_{\text{down}}, b_{\text{up}}$) **to** S
    end if
  end for
  return S
end function

Our parser will be trained and evaluated on monophonic jazz melodies, played by human performers. Monophonic tracks that were not likely to be played by humans were filtered out automatically. This was done by assuming that tracks played by humans contain a lot of variation in onset velocity and in inter-onset intervals caused

by expression and motor noise. From this subset of performed tracks, tracks containing melodies were selected by hand.

After the filtering process, 20 candidate-tracks were left containing unique performances of 12 different melodies. These MIDI files where converted to note lists of the following format:

$$N = [n_0, n_1, \cdots, n_N],$$
$$n_i = (\text{On}_i, \text{Pitch}_i, \text{Velocity}_i),$$

where N is a note list containing notes n_0 to n_N, On_i, Pitch_i and Velocity_i is the onset time in micro seconds, pitch and velocity of the i^{th} note. The following annotation format was chosen: a list of metrical onsets, measured in quarter notes, with pointers to the corresponding notes in N. There are three types of annotations: an onset, a grace note and a rest. Although this study will ignore grace notes and rests, they are included for completeness and potential future use of the corpus.

An annotation A, corresponding to note list N has the following format:

$$A = [a_0, a_1, \cdots, a_N],$$
$$a_j = (\text{Position}_j, \text{Pointer}_j, \text{Type}_j),$$

where Position_j is a metrical position, measured in quarter notes, Pointer_j is a pointer that points to the index of the corresponding note in the note list and Type_j indicates whether this annotation is an onset, a grace note or a rest. Since rests are not included in the note list, their pointer is irrelevant and points to zero.

Some extra information is added to the annotation when it is stored. The annotation A is stored in a 3-tuple (T, t, A), where t is the tempo in beats per minute and T is the time signature. The time signature contains the number of beats per measure, or measure division d and a number u by which we have to divide 1 to get the units of those beats, where u can be any positive real number (although u is almost always a power of 2).

$$T = (d, u)$$

This representation is identical to the musical representation of time signature. The information in the time signature combined with onsets measured in quarter notes can be used to derive the measure number of every onset in the annotation. To do so, the onset in quarter notes, q, is first converted to an onset in beats, b, as follows: $b = q \times u/4$. A position measured in beats can then be converted to a position measured in measures, m, like so: $m = b/d$

(a) Conventional notation. (b) Intended rhythm.

Figure 2.3: Notation of swung notes.

Many performances in the corpus were played in ‘swing’. Although swing may refer to many intentional expressive deviations, a very common one is to play notes that are notated as eighth notes as eighth note triplets, the so-called ‘shuffle’. This is illustrated in figure 2.3. The score usually notates swung notes as in figure 2.3a, whereas the notes are played as in figure 2.3b. Writing swung notes as in 2.3a is just a notational convention, and often the scores contain instructions to play eighth notes as in figure 2.3b.

The manual annotation process resulted in lists of metrical onset times. Combined with the time signature and tempo information, the metrical onset times can be converted to any metrical unit and to metronomic onset times. Deriving te subdivision trees from this information was done semi-automatically. A simple parser with a flat prior and a likelihood function that only allowed metronomic timing was used to generate parses for every item in the corpus. From these results correct parses where selected by hand.

The result of this process for Thelonious Monk’s standard Blue Monk is shown in figure 2.4. Combining pitch information with our subdivision trees allows us to generate scores. Note that our subdivision trees do not represent harmonic information, so even though this transcription should be in the key B-flat, the score does not contain a key signature. A full list of jazz standards in the corpus is given in appendix C.

2.6 Training

The expression-aware model is trained using maximum likelihood estimation. First, for all items in the train set, the observed parameters are and their corresponding feature vectors, the resulting feature vector/parameter pairs (ϕ, p) are stored in a set S. Second, we train two parameter vectors, μ and σ to contain the expected expression

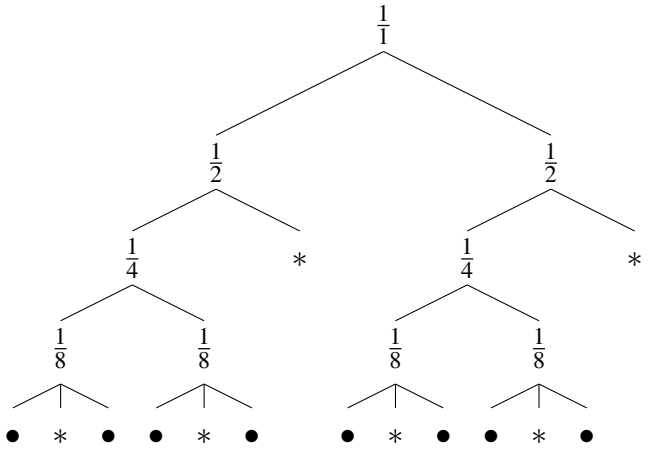
The performance onsets in milliseconds:

$P = [32, 348, 504, 836, 1940, 2240, 2420, 2728]$

Metrical onsets in quarter notes. Triple divisions are rounded to two digits:

$A = [0.0, 0.66, 1.0, 1.66, 4.0, 4.66, 5.0, 5.66]$

Rhythmic analysis generated by a simple parser and selected by hand:



Score generated from the subdivision tree combined with pitch information. The bar duration was set to level 1/2 of the subdivision tree.

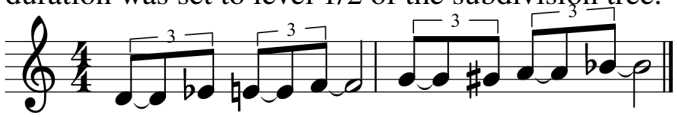


Figure 2.4: A performance of the first two bars of Thelonious Monk's Blue Monk, the corresponding metrical onsets, rhythmic analysis and score generated from the analysis.

ratio r and standard deviation of r for every feature vector ϕ like so:

$$\mu_\phi = \frac{1}{|S|} \sum_{(p|(p,\phi)\in S))} p, \tag{2.12}$$

$$\sigma_\phi = \frac{1}{|S|} \sum_{(p|(p,\phi)\in S)} (\mu - p)^2.$$

Since we use a very simple feature vector, these parameters do not need to be smoothed.

Given the parameter vectors μ and σ, estimated by 2.12, the likelihood of a set of observations S containing feature/expression ratio pairs (φ, r), observed in some hypothesis h is given by:

$$\mathcal{L}(S|\mu,\sigma) \propto \prod_{(\varphi,r)\in S} \exp\left(-\frac{(\mu_\varphi - r)^2}{2\sigma_\varphi^2}\right). \tag{2.13}$$

As mentioned in section 2.2, a hypothesis is rejected if the per-item likelihood is

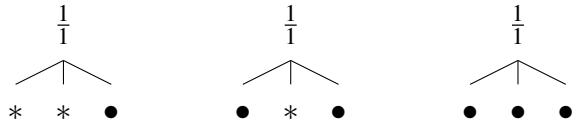


Figure 2.5: Allowed triple divisions.

lower than a certain threshold, the beam parameter. The per-item likelihood is defined as:

$$\mathcal{L}(S|\mu,\sigma) \text{ per item } \propto \exp\left(-\frac{1}{|S|}\sum_{(\varphi,r)\in S}\frac{(\mu_\varphi - r)^2}{2\sigma_\varphi^2}\right). \tag{2.14}$$

The beam parameter is set by taking the minimum value of the per-item likelihood of each subdivision tree in the train set.

Our alternative expression model treats expressive deviation as noise. Deviations from idealised onsets cause expression ratios to be non-zero. The alternative expression model assumes that expression ratios are normally distributed with a mean of zero.

2.7 Implementation

To make the parser computationally tractable, a few optimisations were necessary. It was already mentioned that the parser uses a beam parameter to reject unlikely hypotheses. In addition to that a few extra measures had to be taken.

First of all, the OBSERVATIONS function returns no parameters for hypotheses containing a single onset (complex onsets). Their likelihood is therefore always one and theoretically ties could be added endlessly. Since single-note analyses deeper than a few levels rarely occur, single note hypotheses were allowed to have a maximum depth of five levels.

Second of all, since the corpus contains only 4/4 time signatures, triple divisions are restricted to note level. And of those triple divisions, only the triple divisions shown in figure 2.5 are allowed.

Chapter 3

Evaluation

This chapter will first give an overview of how we will evaluate the parser. The rest of this chapter is structured as follows: An overview of the evaluation criteria is given in section 3.1 and the evaluation measure will be explained in section 3.2.

The parser produces a ranked list of hypotheses. The rhythmic analysis R that corresponds to the most highly ranked hypothesis is assumed to be the parser's interpretation of its input. A parser output R will be evaluated against a gold-standard parse R^* from the corpus by an evaluation measure function.

We cannot afford to keep a completely separate test set since our corpus is fairly small. Therefore we will train the models on different training sets and evaluate them on small parts of the corpus left out of the training set. This process is known as cross-validation. For n-fold cross-validation, the corpus is divided into n equal parts, a training set is constructed from $n-1$ of these parts and a test set is constructed of one of these parts. This is done n times and the parts sampled randomly from the corpus. When dividing into n random sampled parts, performances are treated as whole units. Since there are twenty different performances in the corpus, using 10-fold cross validation results in training sets of eighteen performances that will be evaluated on test sets of two performances.

Furthermore, some performances contain over a hundred onsets. The parser is too slow to parse performances longer than around thirty onsets. Therefore the parser will be tested on small slices of performances, consisting of the first few notes of a performance with some preferred length.

Our subdivision trees can only represent performances with a length, measured in whatever metrical unit, that is a power of two. If a performance is for example three bars long, the subdivision tree will represent a fourth bar as a tie. To make evaluation

(a) Gold-standard. (b) Parser output.

Figure 3.1: An example of incorrect division detection.

simpler, the test performances are restricted to have lengths that are a power of two. This is done by selecting, given some preferred length, the leftmost nested subdivision tree that contains a number of onsets closest to the preferred length. To do this, we use the subdivision trees in our corpus.

3.1 Criteria

To assess the quality of a parser output we will look at how many properties of the gold-standard rhythm were analysed correctly and how many properties of the parse occur in the gold standard rhythm. The subdivision trees describe two properties of rhythm: the location of the down- and upbeats, which is also called the *phase*, and the subdivision pattern, which provides information about the time signature. The evaluation measure should somehow measure the extend to which the parser output is consistent with the gold standard in terms of subdivision and down- and upbeat locations.

A parser output may be consistent with some aspects of the gold-standard but inconsistent with others. In figure 3.1 for example, the parser correctly identified the downbeat but incorrectly assumed a triple division. It is also possible that the parser correctly identifies the divisions but incorrectly identifies the phase, examples of these kind of errors are shown in figure 3.4. Note that a phase error at the deepest level is more severe than a phase error at a higher level. If the phase is incorrect at the deepest level, every downbeat will be identified as an upbeat and vice-versa. If the phase is incorrect at a higher level, the down- and upbeats at the lowest level are still correct. This can be understood intuitively as well: a phase error at the lowest level is more severe since it makes the entire rhythm syncopated, for a phase error at a higher level, the notes on the lower levels would still be in phase.

Notably, figure 3.4 introduces rests in the parse trees. Rests are necessary to represent that the last note is a quarter note and not a whole note. The parser introduced in in this thesis 2 does not handle rests. This issue is discussed further in chapter 5.5.

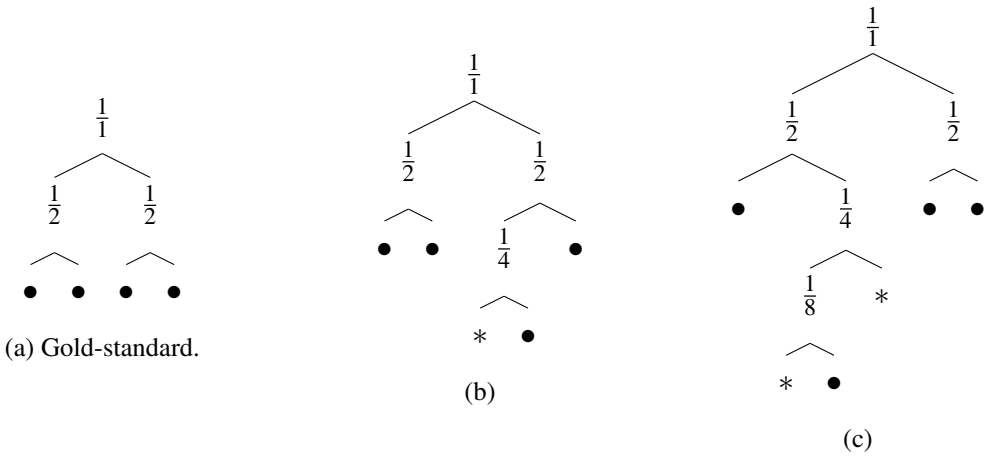


Figure 3.2: An example of too detailed analyses (resulting in a lower precision)

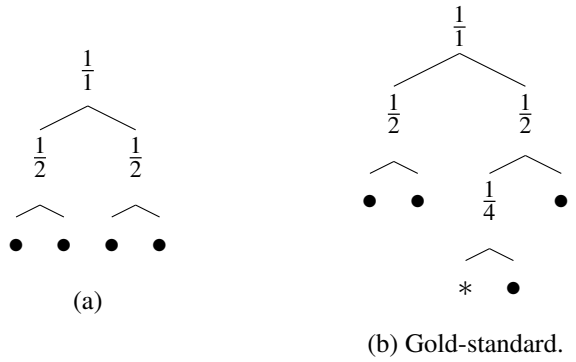
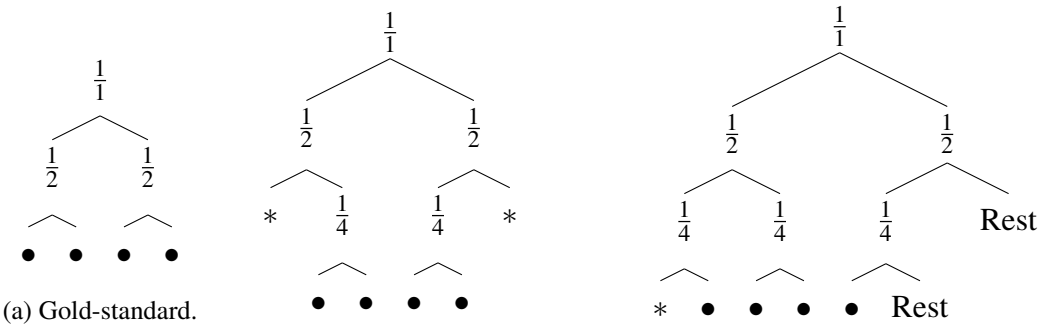


Figure 3.3: An example of too simple analyses (resulting in a lower recall).



(a) Gold-standard.

(b) Phase error at the half note level

(c) Lowest level phase error (most severe).

Figure 3.4: An example incorrect phase detection. See section 5 for why there are rests in this subdivision tree.

3.2 Evaluation Measure

The precision and recall of the parser are measured as follows: The precision is the number of subdivisions and down- and upbeats in the parser output that were correctly identified with respect to the gold-standard, divided by the total number of subdivisions and down- and upbeats in the parser output. The recall is the number of subdivisions and down- and upbeats in the gold-standard that are correctly identified by the parser output, divided by the total amount of subdivisions and down- and upbeats in the gold-standard.

To measure these quantities, every onset governed by R and R^* is converted to a list of claims that this onset makes about the structure. For example, the first quarter note of a 4/4 measure claims to be a downbeat at the half note level and a downbeat at the quarter note level (see for example figure 3.4a). The second quarter note claims to be governed by a downbeat at the half note level and to be an upbeat at the quarter note level. The third onset claims to be an upbeat at the half note level and a downbeat at the quarter note level.

An evaluation that uses these kinds of lists penalises phase errors at the lower levels more severe than phase errors at higher levels. Given the gold-standard in figure 3.4a, consider the following hypothetical errors in the parser output: Should the phase be wrong at the quarter note level (figure 3.4b), a downbeat at the quarter note level becomes an upbeat, however, the downbeat would still be governed by the downbeat at the half note level. Should the phase be wrong at the half note level (figure, 3.4c), the down- and upbeats at the quarter note level are still correct.

There are two issues with the evaluation as suggested above. First, since every onset lists all claims it implies at higher levels, the parser will get points for divisions at higher levels multiple times. Second, figure 3.4 shows that a phase error can lead to another level to be added to the tree, which introduces an inconsistency in the interpretation of levels between the parse tree and the gold-standard.

The first issue is remedied by keeping a list of parser decisions that have been accounted for. If the first onset in figure 3.4a claims a downbeat at the half note level and a downbeat at the quarter note level, both of these claims are added to a list of claimed facts. The second onset claims an upbeat at the quarter note level and a downbeat at the half note level. Since the downbeat at the half note level is already the list of claimed facts it will not be counted again.

To reiterate the first paragraph of this section more concretely, we define the evaluation function $\text{SCORE}(R, R')$ which counts the number of claims that are shared by R and R' and divides that by the total number of claims in R. The precision is defined as the number of claims in R that appear in the claims of R^* as well, divided by the total number of claims in R, so:

$$\text{precision} = \text{SCORE}(R, R^*). \tag{3.1}$$

Similarly, the recall is defined as:

$$\text{recall} = \text{SCORE}(R^*, R). \tag{3.2}$$

Consider the example in figure 3.2. The current evaluation assigns a precision of $\frac{6}{7}$ and a recall of $\frac{6}{6}$ to the parse in figure 3.2a. This implies that a total of 6 claims are made by the gold-standard. These are: the four down- and upbeats at the quarter note level and one downbeat and one upbeat at the half note level. The parse in figure 3.2b makes the same claims but also claims an upbeat at the eighth note level, which is incorrect.

We have not addressed the second issue yet. That is, we are not sure if the top level of R corresponds to the top level of R^*. Either of those levels may have been added as a result of a different phase in R and R^*. The only solution seems to be to consider three scenarios, illustrated by figure 3.5: (1) No extra levels have been added and the levels in R are consistent with the levels in R^*. (2) Let R^* be figure 3.5a and R be figure 3.5c, then a phase error resulted in an extra level in R. (3) Let R^* be figure 3.5c and R be figure 3.4a, then a phase error resulted in one level less in R.

If we want to get a valid precision and recall in scenario (2), we should convert R to the tree in figure 3.5b by adding a top-level tie. In scenario (3) we want to convert

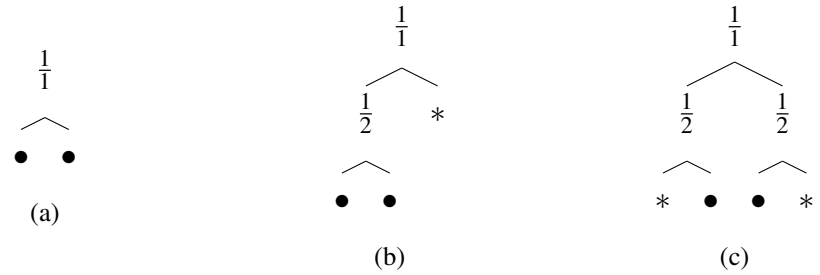


Figure 3.5: Extra levels added by phase errors

R^* to the tree in figure 3.5b by adding a top-level tie. This suggests three scenarios for evaluation: (1) evaluate R against R^*, (2) evaluate $(R,*)$ against R^* and (3) evaluate R against $(R^*,*)$.

The evaluation function in its final form is:

$$\text{precision} = \text{MAX}(\text{SCORE}(R,R^*),\text{SCORE}((R,*),R^*),\text{SCORE}(R,(R^*,*))), \tag{3.3}$$

$$\text{recall} = \text{MAX}(\text{SCORE}(R^*,R),\text{SCORE}((R^*,*),R),\text{SCORE}(R^*,(R,*))). \tag{3.4}$$

Chapter 4

Experiments and Results

First, we will train the rhythm and expression model on the entire jazz corpus as we described in section 2.6 and analyse the results. Second, we will use 10-fold cross validation to evaluate the performance of the parser on the corpus. We will compare the expression model presented here to an expression model that treats expression as additive noise. That is, for any feature vector μ_φ is set to zero and σ_φ is set to a small value.

4.1 Rhythm and Expression Model

The rhythm and expression model, trained on the entire jazz corpus are shown in table 4.1.

The expression model shows a slight stretching of downbeats in duple divided units at the lowest level. The values shown in 4.1b are logarithmic ratios. Therefore, the average stretching of downbeats at level one can be found by taking the exponential: $\exp(7.926x10^{-2}) \approx 1.082$.

The small μ and σ values at higher levels are a consequence of the way that the performers of our corpus recorded the melodies. As we have said earlier, the expression ratio reflects tempo changes at higher levels. The melodies in the jazz corpus were often played along with a metronomic accompaniment track so that the global tempo does not change. This results in near-zero expression ratios at high levels.

Table 4.1: The rhythm and expression model, trained on the entire jazz corpus.

(a) The rhythm model.

Rule	**Probability**
$R \rightarrow \bullet\ \bullet$	0.061
$R \rightarrow *\ \bullet$	0.027
$R \rightarrow R\ R$	0.392
$R \rightarrow \bullet\ R$	0.057
$R \rightarrow R\ \bullet$	0.022
$R \rightarrow *\ R$	0.069
$R \rightarrow R\ *$	0.061
$R \rightarrow \bullet\ \bullet\ \bullet$	0.013
$R \rightarrow \bullet\ *\ \bullet$	0.164
$R \rightarrow *\ *\ \bullet$	0.134

(b) The expression model.

$\varphi = [\texttt{level},\texttt{division}]$	μ_φ	σ_φ
$[1,2]$	$7.926x10^{-2}$	$3.391x10^{-2}$
$[1,3]$	$-6.323x10^{-2}$	0.656
$[2,2]$	$-4.68x10^{-3}$	$2.184x10^{-2}$
$[3,2]$	$5.565x10^{-3}$	$9.422x10^{-3}$
$[4,2]$	$4.797x10^{-3}$	$9.824x10^{-3}$
$[5,2]$	$-3.391x10^{-3}$	$1.887x10^{-2}$
$[6,2]$	$-7.539x10^{-5}$	$1.375x10^{-3}$
$[7,2]$	$-8.029x10^{-3}$	$1.223x10^{-3}$
$[8,2]$	0.0	0.0
$[9,2]$	0.0	0.0

4.2 Performance on the Corpus

We prepared three experiments which we tested on four different preferred lengths: 5, 10, 15 and 20 (see chapter 3). In the first experiment we used our PCFG rhythm model in combination with our expression model, the results are shown in table 4.2. In the second experiment, we used our own PCFG rhythm model combined with the alternative expression model that treats non-zero expression ratios as noise with a standard deviation of 0.1. The results of this experiment are shown in table 4.3. The final experiment served as a baseline: a rhythm prior that assigns the same probability to every rhythm was used in combination with the expression model that treats expression as noise. The results of the final experiment are shown in table 4.4.

Table 4.2: Results for the expression model with the PCFG prior.

Preferred length	**Precision**	**Recall**	**F-score**
5	0.486	0.464	0.475
10	0.484	0.530	0.506
15	0.517	0.568	0.541
20	0.498	0.525	0.511

Table 4.3: Results for the model that treats expression as additive noise with $\mu = 0$ and $\sigma = 0.1$ and the PCFG prior.

Preferred length	**Precision**	**Recall**	**F-score**
5	0.842	0.789	0.815
10	0.718	0.692	0.705
15	0.850	0.831	0.840
20	0.682	0.696	0.689

Table 4.4: Baseline results. Expression is treated as additive noise and the prior returns the same probability for every rhythm.

Preferred length	**Precision**	**Recall**	**F-score**
5	0.378	0.518	0.437
10	0.410	0.55	0.470
15	0.396	0.472	0.431
20	0.389	0.471	0.426

Chapter 5

Discussion

This chapter will discuss he effectiveness of the PCFG prior, our expression models and the extent to which our results support the claims we have made in chapter 1.

5.1 Subdivision Parsing

Apart from the results for our expression-aware model, which we will discuss in section 5.3, our results for the alternative expression model seemed satisfactory and were significantly higher than the baseline. In general, this shows that subdivision parsing has been a successful approach. In the future, it would be interesting to see how our parser performs on other corpora and compare its performance directly to other models of rhythmic structure perception.

The first thing to note about the expression-aware model and the alternative expression model is that they are both based on expression ratios. Although the alternative expression model treats expressive deviation as noise, it does not penalise performances for not sticking to one tempo; it only penalises performances for changing tempo. For example, a performance that is completely metronomic except for slowing down halfway would cause high-level expression ratios to be non-zero while low-level expression ratios remain zero. Likewise, the stretching of a downbeat somewhere causes only one low-level expression ratio to be non-zero instead of causing every note afterwards to be off with regard to some metronomic timing, as might happen in a model that assumes notions of tempo. This illustrates the natural way in which our tempo-independent approach treats expression.

Figure 5.1a and 5.1b illustrate how the parser interpreted the rhythmic structure of an onset pattern almost completely correct. By combining pitch information with the

parser's analysis and setting a bar's duration to correspond to level 1/8 in the tree, we can generate a score of the onset pattern. This is illustrated in figure 5.1c.[1]

To generate a score from a subdivision tree we need to set two variables by hand: the time signature and the bar duration. This information cannot be trivially deduced from the subdivision tree. Although the subdivision tree does specify whether beats are duple or triple divided, it does not differentiate between time signatures like 4/4, 2/2 and 2/4. Which one of these is preferred may depend on how beats are being stressed.

However, the problem of determining the duration of a measure may be solved quite easily. We could use a simple model that tries to find a tactus level in the tree, preferring intervals some interval that have been found to correspond to the tactus level. We can derive this parameter from our corpus as we know the metrical onset time of each note in our subdivision trees of performances.

Setting a tactus level to some preferred interval is similar to the systems in Temperley (2009, 2007). The difference is that in those models, finding the tactus level is the first step in constructing the rhythmic structure. We reverse the order: first, we construct the rhythmic structure, then we find the tactus level to derive the correct bar duration. For the tree in figure 5.1b for example, we might find that the 1/32 level corresponds most closely to some preferred tactus interval.

As mentioned earlier, the parser's interpretation in figure 5.1c differed slightly from the gold-standard in figure 5.1c. The difference is lies in the way the parser interpreted the first onset. The gold-standard specifies that the first bar should be divided into two half notes, the first half note is divided in two quarter notes and the rightmost of these quarter notes is divided into an eighth note triplet with an onset on the last position. Instead of this rather complicated analysis, the parser simply divided the bar in a half note triplet with an onset on the last position.

Further inspection of the two interpretations reveals that they are interchangeable. Triple divisions introduce an ambiguity in subdivision trees, illustrated in figure 5.2. The parser chose the simplest interpretation since it has the highest prior probability. The difference between the two analyses can be expressed as a difference in time signature. The analysis on the left in figure 5.2 implies a 3/2 time signature, the analysis on the right implies a 4/4 time signature where the third beat is divided into three eighth notes.

The gold-standard in figure 5.1d was chosen because there is a certain consistency

[1] Our system is incomplete as a music transcription system: scores generated from subdivision trees do not contain key signatures since our system does not include harmonic analysis.

of time signature: we assume the time signature not to change unless the performance gives us clear evidence that it should change. The rhythm model as it was presented here has does not penalise changes in time signature and adding this could be a potential improvement to the model.

5.2 The Rhythm Model

An appropriate evaluation of our rhythm model would be to compare its performance to Temperley's hierarchical model, used in (Temperley, 2009). We could for example exchange our prior for Temperley's and compare the parser's performance on our corpus. We did not implement this comparison at present for two reasons: First, Temperley did not specify how his hierarchical model generalises to triple divisions. Second, the hierarchical model is defined relative to the levels of a metrical grid. Levels in our model do not correspond trivially to levels in a metrical grid.

Although we cannot compare the PCFG prior to the hierarchical model directly, we can make some observations that show that the PCFG prior, even when trained on a jazz corpus, seems to reflect some notions of what Temperley (2010) calls common-practice rhythm.

Looking at the results in table 4.1a, we can easily see that the prior penalises syncopation: the probability of $R \rightarrow * \bullet$ is lower than the probability of $R \rightarrow \bullet \bullet$ and the probability of $R \rightarrow \bullet * \bullet$ is lower than the probability of $R \rightarrow * * \bullet$.

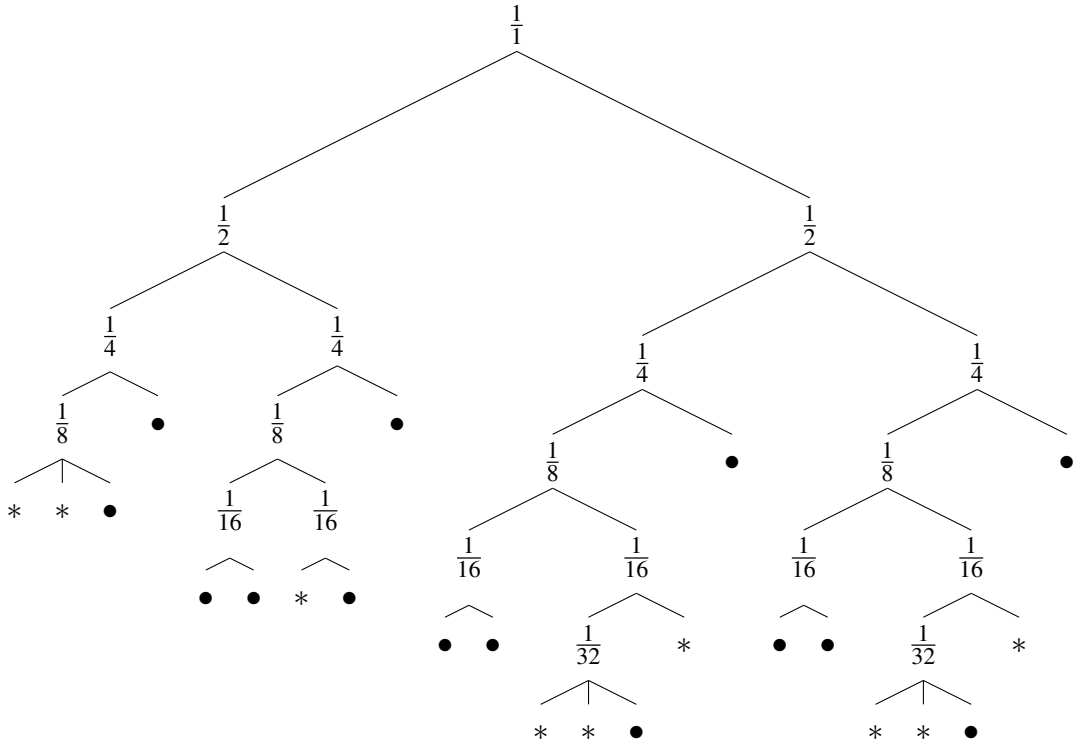
It is also clear that simpler analyses are preferred to more complicated analyses, reflecting that onsets on deep levels of subdivision trees are less likely. In the case of the two ambiguous analyses in figure 5.2 the simpler one will always be preferred by the model since the probability of of some analysis R is constructed as the product of the probabilities of the rules applied to construct R. The simpler analysis contains less rule applications and will therefore be preferred.

It is less clear that long notes on downbeats are preferred but we can observe that the probability of $R \rightarrow R \bullet$ is smaller than the probability of $R \rightarrow R \bullet R \rightarrow \bullet R$. This may indicate a preference for long notes on downbeats.

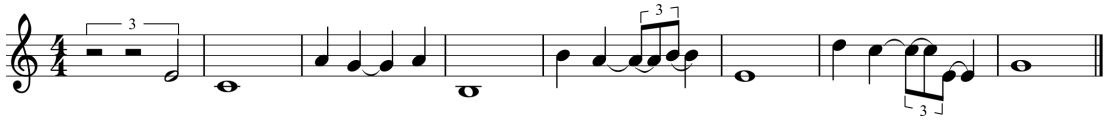
Figure 5.3 shows an example that Temperley (2010) uses to introduce his hierarchical model. The hierarchical model classifies the rhythm on the left as more likely (that is, more common-practice) than the rhythm on the left because eighth note upbeat is followed by a quarter note downbeat. The PCFG prior gives us the same result: With respect to the tree on the left, the tree on the right has replaced the rule expansion

[2.62, 3.03, 4.52, 4.92, 5.65, 6.02, 7.54, 7.91, 8.53, 9.04, 10.51, 10.89, 11.53, 12.05]

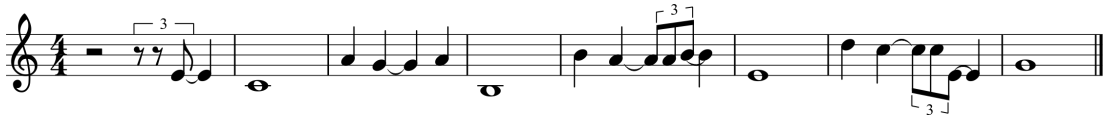
(a) The raw onset times of the performance (in seconds).



(b) The interpretation generated by the parser.



(c) A score generated from the subdivision tree combined with pitch information. The bar duration was set to correspond to the 1/4 level in the tree.



(d) The gold-standard score.

Figure 5.1: A parser interpretation that was almost the same as the gold-standard.

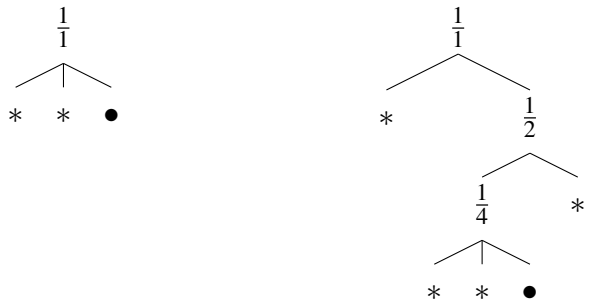


Figure 5.2: An ambiguity introduced by triple divisions.

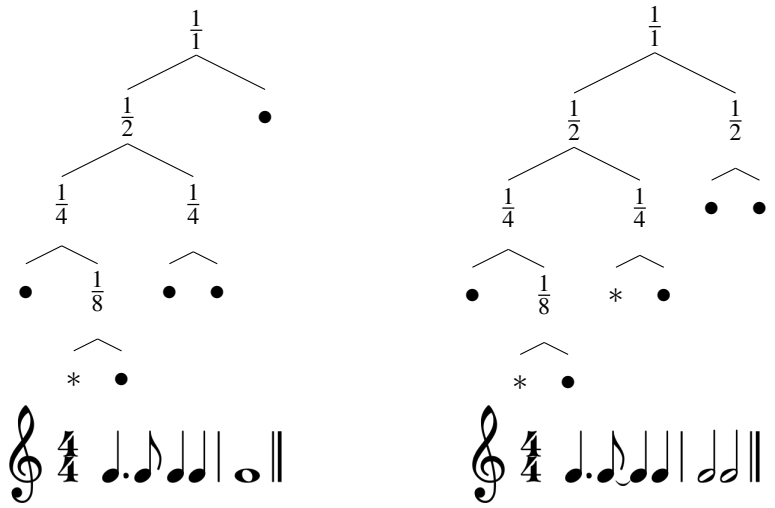


Figure 5.3: Two examples from Temperley (2010) and their subdivision trees.

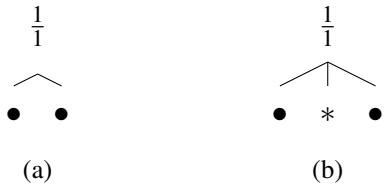


Figure 5.4: Two competing interpretations of two onsets.

$R \rightarrow \bullet\ \bullet$ at level 1/4 with $R \rightarrow *\ \bullet$, which indicates syncopation. This is in theory the same principle as the hierarchical model captures: an upbeat that is not preceded by a downbeat results in $R \rightarrow *\ \bullet$ rule expansions, which are less likely that $R \rightarrow \bullet\ \bullet$, at least in our model trained on a jazz corpus.

The PCFG prior thus seems to be an adequate model of rhythm and our results show that some properties of rhythm that Temperley (2010) identifies as common-practice indeed generalise to some extent to jazz music.

5.3 The Expression Model

The expression model that treats non-zero expression ratios as noise performed better than our baseline. The expression-aware model, however, did not perform significantly better than our baseline. In this section, we will discuss the expression model and offer some explanations for the low evaluation scores of the expression-aware model.

A high amount of noise level-one triple divisions seems to be the biggest contribution to the low evaluation scores. The high value of the σ parameter for level-one triple divisons, combined with a prior that assigns a high probability to the rule $R \rightarrow \bullet\ *\ \bullet$, divisions like in figure 5.4a are likely to be classified as the division in figure 5.4b. Because of the high σ for triple divisions at level one, the incorrect analysis in 5.4b is not penalised very much and the high prior probability of analysis in figure 5.4b may make the parser prefer it to the analysis in figure 5.4a. The additive noise expression model sets σ to be 0.1 for all levels and divisions and penalises the interpretation 5.4b of 5.4a more severely.

This interpretation is supported by some of the parses that the parser produced where duple divisions are interpreted as triple divisions. Figure 5.5 shows the parser's interpretation of the first four measures of Kenny Dorham's Blue Bossa. The parser's interpretation is sometimes quite at odds with the gold-standard, but given that the incorrectly interpreted divisions are all triple divisions at the lowest level, this parse is

not penalised much for it.

It seems thus that the expression model has a consistent bias towards classifying divisions as triple divisions. This may explain why did not perform much better than the baseline.

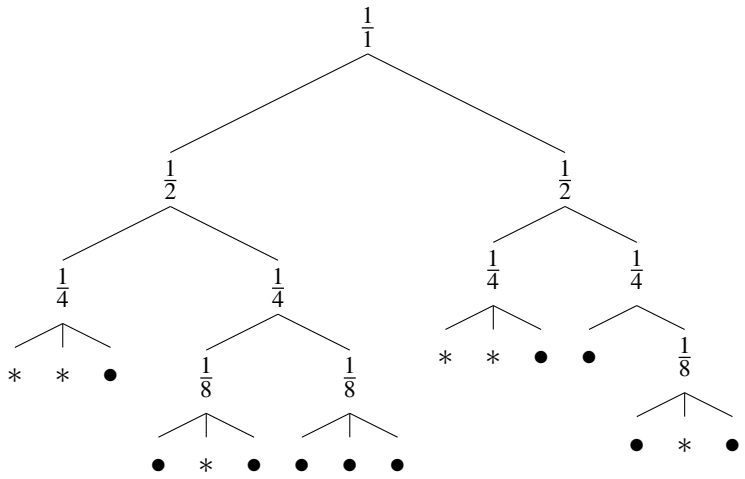
The down-/upbeat for divisions higher than two is calculated by averaging the beat lengths before the onset and dividing it by the average beat length after the onset (see equation 2.9). We do not use a feature for different kind of triple divisions but average all triple divisions together during training. One drawback of this approach is that the swing ratio is lost.

The question remains why the expression model is so noisy for triple divisions. A possible reason is that swung notes are not consistently played in a 2:1 ratio, as our annotated corpus assumes. This is consistent with findings that suggest that swing ratio scales with tempo in a non-linear way (Honing and Haas, 2008).

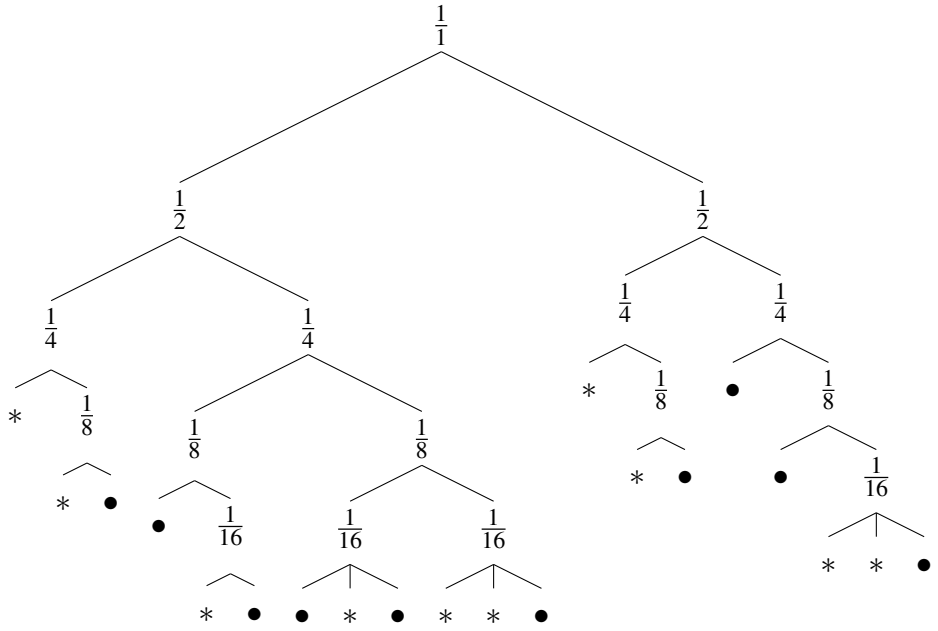
Yet, it seems that the σ parameter of roughly 0.7 cannot be caused only by variation in swing ratios. The exponential of 0.7 is approximately a ratio of 2, which seems unlikely to happen very often when performers are reasonably competent. After analysing the corpus, it was found that out of all 959 expression ratios at a level-one triple division, 81 had an expression ratio of higher than $\log(2)$ or lower than $\log(0.5)$. A few of those may have been caused by annotation errors but it seems that most of these are caused by the way the ratios are observed by the OBSERVATIONS function.

The top-down OBSERVATIONS function, however, partially relies on expected onsets calculated by the bottom-up BEATS function. It is hard to determine how exactly the rules and assumptions in both of these functions and affect the results. The interpretation of our results would benefit from further formalisation of these functions.

A final potential factor influencing the results is that the level feature is not completely consistent. In section 2.4, the `level` feature was defined as the depth of the hypothesis at which a expression ratio is observed. However the lowest level of a tree is not always the same level. Level one for example can be the quarter note level at some point in a performance and the eighth note level at another point in the same performance. Parameters of the expression model for level one may have been trained on expression ratios observed in eighth notes and quarter notes and other notes that happened to the lowest level in span of a performance. It seems that this may have negatively influenced the scores of our expression model.



(a) Parser interpretation



(b) Gold-standard

Figure 5.5: The parser interpretation and gold-standard of the first four measures of Kenny Dorham's Blue Bossa.

5.4 The Jazz Corpus

We have pointed out earlier that our jazz corpus contains melodies that were accompanied by metronomic tracks, resulting in very low deviation in expression ratios at higher levels. Extracting melodies from polyphonic tracks produced another interesting issue.

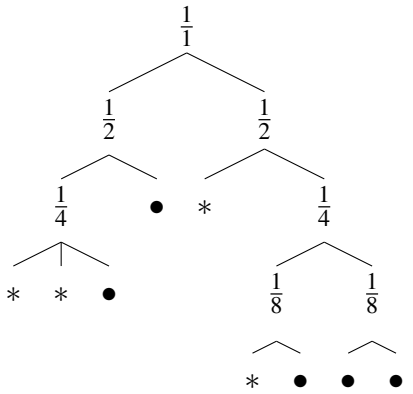
Occasionally, the parser produces interpretations that may have been more sensible to the listener than the gold-standard interpretation of a rhythm. An example of this is shown in figure 5.6, which shows the rhythm of the first four measures of Chick Corea's Brazil. The gold-standard analysis claims that the first two beats are swung eighth-note upbeats. This is a very odd way for a piece to start. The parser makes the much more sensible suggestion that the second onset is actually a downbeat on the first beat of the second measure. The second onset is only a third of a quarter note away from the downbeat of the second measure so the parser's analysis is not very far off.

The reason why the performance in the corpus oddly starts with two upbeats followed by a measure of silence is probably that it originally was accompanied by other instruments, in which case it would not be as unusual to start a melody like this. Since we are considering the melodies in isolation and we test on short fragments, we occasionally get melodies where it is, even to human listeners, not immediately clear where the downbeats are. The way the parser annotated this rhythm may well be the way that most listeners would interpret it. In any case, the unconventional beginning of this melody seems to be a drawback of having a monophonic jazz corpus constructed from polyphonic MIDI files.

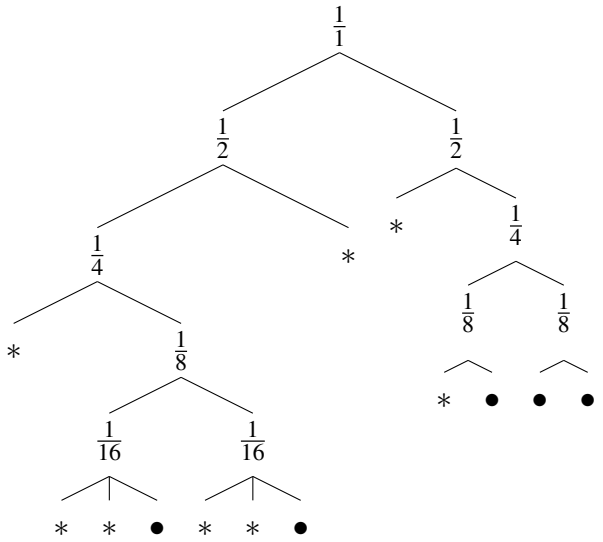
5.5 Rests

We assumed early on that we could completely represent the rhythmic structure by looking only at onset times. It the evaluation section we referred to this section when we needed rests in the parse tree to represent a quarter note at the end of the performance. It turns out that, while we can completely specify the rhythmic structure for the most part using just onsets, we need rests to represent a performance that does not end on the last beat of a bar.

Consider the gold-standard in in figure 5.7a and a potential parser interpretation in figure 5.7c. Figure 5.7c implies that the last note, which is a quarter note in the gold-standard, is a whole note like in score B in 5.7d. Another interpretation is that



(a) Interpretation produced by the parser with the additive noise expression model.

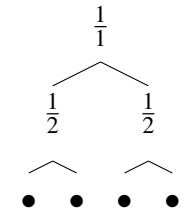


(b) Gold-standard.

Figure 5.6: A comparison of the parser and gold-standard interpretation of the first four measures of Chick Corea's Brazil.

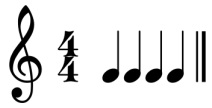
the last note is a half note and the others eighth notes. Both of these interpretations are quite unlikely. The parser however, considers only the onset of the last note and has no way of knowing how long the duration of the note is, so the tree in figure 5.7c is not as unlikely as it should be. Even if the parser knows the duration of the last note, then given the current grammar, it has no way of representing the last note as a quarter note.

The only solution is to extend our approach to include rests. To do so would not require a great deal of changes. We can use roughly the same approach and add a *rest* unit to the grammar, whose onset can be considered to be the offset of the previous note, tie or rest. In such an approach, rests would simply be another unit that have an onset which can be compared to its expected onset. This, however, may data sparsity a greater problem and may require a larger corpus than we used here.

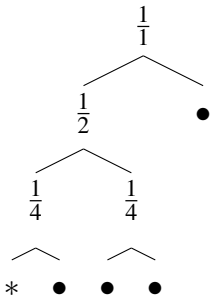


(a) Gold-standard.

A

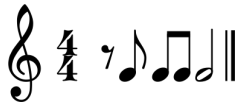


(b)

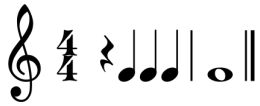


(c) Parser interpretation.

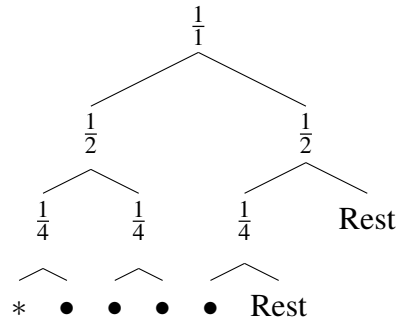
A



B

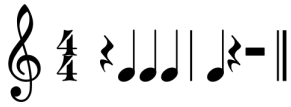


(d)



(e) A representation that includes rests.

A



(f)

Figure 5.7: An example of why rests are desirable to correctly represent the last note.

Chapter 6

Conclusion

We proposed a system that, in the first place, was aimed at interpreting rhythmic structure in performances. We proposed an approach that combines chart-parsing with a Bayesian model of rhythmic structures. This model consisted of a rhythm model and an expression model. The rhythm model was a probabilistic context-free grammar that assigned probabilities to context-free rule expansions of subdivision trees. Two expression models were proposed, both based on down-/upbeat length ratios. The expression-aware model could potentially learn expressive deviation biases, linked to rhythmic structure in performances, the alternative expression model treated expression, defined as down-/upbeat ratios non-equal to one, as noise.

Furthermore, we produced an annotated corpus of monophonic performances of jazz melodies. The performances are annotated with metrical onset times, time signatures, and rhythmic structure represented as subdivision trees. Since there are few other corpora with this kind of information and, as far as we know, it is the first of its kind that focusses on jazz music, this corpus could potentially be useful in future studies in this field.

At the moment our approach only covers time signatures with duple divisions and triple divisions, where triple divisions are restricted to the note level. More work is needed to extend the parser to allow other divisions and triple divisions at higher levels.

Our rhythm model became sensitive to some regularities in rhythms that have been suggested by Temperley (2010) as common practice. This shows that our rhythm model is both an acceptable model of rhythm perception and that common practice rhythm principles generalise to some extent to rhythms jazz music.

We initially claimed that some local expressive deviations correlated with rhythmic structure, as observed by Bengtsson and Gabrielsson (1983), may improve parser per-

formance. Our results, however, showed that the expression-aware model in its current form did not help parser performance. In our discussion of the results we suggested that this was caused by a high amount of noise in the model, probably caused by the way we predict downbeat and next downbeat onsets. At this point, the expression model needs improvement.

Our alternative expression model, which treats expressive deviation as noise, did perform significantly better than the baseline and often produced correct rhythmic structures. This model, like the expression-aware model, treats expression in a natural, tempo-independent way. We have argued that for this reason, subdivision parsing is an elegant approach and that it is particularly well-suited for studying expression. Our results show that a subdivision-based parser is indeed able to produce correct subdivision trees of expressively performed rhythms in jazz music.

Appendix A

Chart parsing rhythms

Modifications to a standard CKY parser were necessary to allow rejection of unlikely hypotheses, to allow hypothesis representations to be build instead of just syntax trees and to deal with the fact that the number of ties is the input performance is unknown.

The parser consists of three functions: PARSE(P, n, b), CLOSE(H, b) and GROUP(H). n is the first beam parameter which specifies the maximum number of hypotheses per cell, b is the second beam parameter which specifies the minimum per observation likelihood for a hypothesis to be included in a cell.

The parser input is a performance P, as defined in section 2. A single onset is referred to as P_i. We write a sub-span from onset i to onset j as P_{ij}, where

$$P_{ij} = [\mathrm{On}_i, \mathrm{On}_{i+1}, \cdots, \mathrm{On}_j]. \tag{A.1}$$

The PARSE(P) function is a largely unmodified CKY algorithm. The only modification is that after a cell has been filled with hypotheses, they are ranked by their posterior probability and everything not in the top-n is rejected using the REJECT(Cell) function.

The CLOSE(H, b) function takes a set of one or two hypotheses H and keeps applying rules in the CFG in 2.1 until no more rules can be applied. The resulting hypotheses that have a per-observation likelihood higher than b are added to a list which is returned when no more hypotheses could be added.

The GROUP(H) implements the CFG rules and constraints 2.1. It takes a list of hypotheses H and returns a list of hypotheses that could be created by combining hypotheses in H. Since we do not know where and how many ties there are in our input, the GROUP function includes hypotheses where ties have been added in its results.

```
function PARSE(P, n, b)
    n ← LENGTH(P)
    t ← n by n+1 dimensional chart
    for j ← 1, n+1 do
        append P_{j-1} to t[j-1, j]
        append CLOSE(P_{j-1}, b) to t[j-1, j]
        for i ← j-2, -1 do
            Cell ← ∅
            for k ← i+1, j do
                for B ← t[i, k] do
                    for C ← t[k, j] do
                        append CLOSE([B, C], b) to Cell
                    end for
                end for
            end for
            Cell ← REJECT(Cell)
            t[i, j] ← Cell
        end for
    end for
    return t
end function
```

```
function CLOSE(H, b)
    Cell ← ∅
    Unseen ← ∅
    while true do hypotheses ← GROUP(H)
        for h ← H do
            O ← OBSERVATIONS(h)
            ℒ ← PER_OBSERVATION_LIKELIHOOD(O)
            p ← PRIOR(h)
            if ℒ > b then
                append (h, p ∗ ℒ) to Unseen
            end if
            if Unseen = ∅ then break
            end if
            append Unseen to Cell
        end for
    end while
    return Cell
end function
```

```
function GROUP(H)
    Result ← ∅
    if LENGTH(H) = 1 then
        h ← H_0
        if DIVISION(h) = 0 then
            append [(∗, h)] to Result
        else
            append [(∗, h), (h, ∗)] to Result
        end if
    else
        h ← H
        append [h] to Result
    end if
    return Result
end function
```

Appendix B

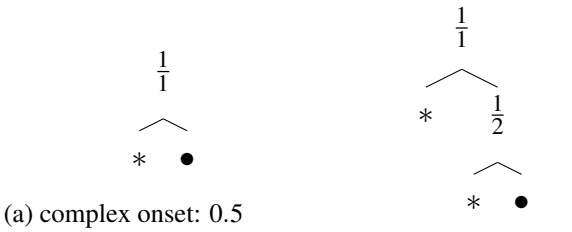
Predicting Onsets

The goal of the BEATS function is to get the best possible estimates of the onsets of down- and upbeats in hypotheses where these are filled with ties. This is a bottom-up recursive process.

The algorithm works roughly as follows: The input is a hypothesis h. First, the algorithm iterates depth-first through the tree by calling the BEATS function for each of its children. When the beats function is called for a single tie or onset (DIVISION(h) = 0), the function returns $*$ or the onset time.

The algorithm keeps a list of onsets and their relative position, from these intervals, the algorithm will derive any missing onset times. Relative positions are the position of a beat on which an onset occurs. If an onset occurs on the downbeat for example, its relative position is 0, if it occurs on the first upbeat, its relative position is 1.

If no onset occurs on a beat, the beat may still *govern* an onset. Such an onset will be treated as a *complex onset* and its position as a *complex position*. For some hypothesis h, governing a single onset, the complex position is derived by the function COMPLEXPOSITION. The hypothesis in figure B.1a for example is treated as an onset



(a) complex onset: 0.5

(b) complex onset: 0.75

Figure B.1: Complex onsets (single-onset analyses).

at relative position 0.5, the hypothesis in figure B.1b is treated as an onset with relative position 0.75.

The algorithm prefers to derive missing onsets from beat intervals of relative positions, but if these are not available, the complex positions will be used. The FILL(onsets, *d*) function derives the expected onsets of every beat in a hypothesis given a list of (complex) positions and onsets.

This process assures that any hypothesis that governs more than one onset will have estimated onset times for each beat. The full algorithm is given in algorithm 2

Algorithm 2 Estimate down- and upbeat onsets given some hypothesis h.

function BEATS(h)
  $d \leftarrow$ DIVISION(h)
  if d = 0 **then**
    return h
  end if
  positions $\leftarrow \emptyset$
  complexPositions $\leftarrow \emptyset$
  for $i \leftarrow 0, d$ **do**
    $B \leftarrow$ BEATS(h_i)
    $O \leftarrow$ ONSETS(h_i)
    if DIVISION(h_i) $= 0$ **and** $O \neq [*]$ **then**
      append (i, O_i) **to** positions
    else
      if $B_0 \neq *$ **then**
        append (i, B_0) **to** positions
        if $O_0 \neq *$ **then**
          append (i, O_0) **to** O
        end if
      else
        p, complexPosition $\leftarrow$ COMPLEXPOSITION(h_i)
        append $(i + p$, complexPosition) **to** complexPositions
      end if
    end if
  end for
  if LENGTH(positions) ≤ 1 **then**
    append complexPositions **to** positions
  end if
  return FILL(positions, d)
end function

Appendix C

The Jazz Corpus

The corpus consists of the following Jazz standards:

1. A Fine Romance (2x)
2. Afternoon in Paris (2x)
3. Ain't Misbehavin' (2x)
4. All the things you are
5. Au privave (2x)
6. Blue Bossa (2x)
7. Blue Monk
8. Blues for Alice
9. Brazil (4x)
10. Daahoud
11. Don't Get Around Much Anymore
12. Everything Happens to Me

Bibliography

Ingmar Bengtsson and Alf Gabrielsson. Analysis and synthesis of musical rhythm. *Studies of music performance*, 39:27–60, 1983.

Ali Taylan Cemgil, Peter Desain, and Bert Kappen. Rhythm quantization for transcription. *Computer Music Journal*, 24(2):60–76, 2000.

Peter Desain and Henkjan Honing. Tempo curves considered harmful. *Contemporary Music Review*, 7(2):123–138, 1993.

Sebastian Flossmann, Werner Goebl, Maarten Grachten, Bernhard Niedermayer, and Gerhard Widmer. The magaloff project: An interim report. *Journal of New Music Research*, 39(4):363–377, 2010.

Mark Granroth-Wilding. *Harmonic Analysis of Music using Combinatory Categorial Grammar*. PhD thesis, University of Edinburgh, forthcoming.

Mitsuyo Hashida, Toshie Matsui., and Haruhiro Katayose. A new music database describing deviation information of performance expressions. In *Proc. of the 9th International Conference on Music Information Retrieval (ISMIR). Philadelphia*, 2008.

Henkjan Honing and W. Bas De Haas. Swing Once More: Relating Timing and Tempo in Expert Jazz Drumming. *Music Perception*, 25(5):471–476, 2008.

Fred Lerdahl and Ray Jackendoff. *A generative theory of tonal music*. The MIT Press, 1983.

Hugh Christopher Longuet-Higgins. Perception of melodies. *Nature*, 1976.

Caroline Palmer. Mapping musical thought to musical performance. *Journal of experimental psychology: human perception and performance*, 15(2):331, 1989.

Christopher Raphael. A hybrid graphical model for rhythmic parsing. *Artificial Intelligence*, 137(1):217–238, 2002.

David Temperley. *Music and probability*. MIT Press, 2007.

David Temperley. A unified probabilistic model for polyphonic music analysis. *Journal of New Music Research*, 38(1):3–18, 2009.

David Temperley. Modeling common-practice rhythm. *Music Perception*, 27:355–376, 2010.

Daniel H. Younger. Recognition and parsing of context-free languages in time n3. *Information and Control*, 10(2):189 – 208, 1967. ISSN 0019-9958. doi: 10.1016/S0019-9958(67)80007-X.