RULE MINING FOR LOCAL BOUNDARY DETECTION IN MELODIES

Peter van Kranenburg
Meertens Instituut, Utrecht University
peter.van.kranenburg@meertens.knaw.nl

ABSTRACT

The task of melodic segmentation is a long-standing MIR task that has not yet been solved. In this paper, a rule mining algorithm is employed to find rule sets that classify notes within their local context as phrase boundaries. Both the discovered rule set and a Random Forest Classifier trained on the same data set outperform previous methods on the task of melodic segmentation of melodies from the Essen Folk Song Collection, the Meertens Tune Collections, and the set of Bach Chorales. By inspecting the rules, some important clues are revealed about what constitutes a melodic phrase boundary, notably a prevalence of rhythm features over pitch features.

1. INTRODUCTION

Melody is one of the basic aspects of music. As such, it has been the object of study in numerous research projects in various fields, including music theory, ethnomusicology, music cognition, and music information retrieval. In virtually all those studies, it is generally accepted that a given melody can be analysed in terms of smaller constituents. The availability of a musically sensible segmentation facilitates various music information retrieval tasks [1]. There is, however, no coherent theoretical answer to the questions what exactly are these constituents and how to isolate them from the holistic construct of a melody.

One line of research has been to design computational models to detect segment boundaries at the surface level of the melody. Typically, these models have been tested on a corpus of melodies in which segment boundaries are annotated, mainly the Essen Folk Song Collection [2].

Computational models that have been proposed to partition a melody into a sequence of segments, basically take one of two approaches. In the first approach, which is theory-driven, a set of rules is designed based on theories of human perception and cognition of melodic information, typically drawing on a combination of Gestalt Psychology [3] and Music Theory. These rules are then formalised and quantised in such a way that they can be implemented in software to automatically detect possible segment boundaries in the melodies. The underlying assumption is that these rules reflect the way humans detect patterns in sensory input.

In the other approach, which is data-driven, a model is learnt from data without strong a-priori theoretic assumptions. This approach is based on the idea that a human listener learns to recognise musical events (such as segment boundaries) by exposure.

In this article, we apply a rule mining algorithm that infers from a large corpus of segmented melodies a rule-based model of what is a phrase boundary. The choice for rule mining is motivated by the explainability of the resulting models, which consist of human readable sets of rules. By examining the discovered rules we gain a better understanding of what constitutes a melodic segment boundary, and what features play a role for detecting segment boundaries. We include many features to allow the mining algorithm to choose which features are necessary for the task. We apply the rule-mining algorithm RIPPER [27] as wel as a Random Forest classifier [29] on several subsets of features. By using other data sets next to the Essen Folk Song Collection, we broaden the information on which the models are based, and we are able to compare phrase boundaries across different melodic styles.

2. RELATED WORK

In this section, we review relevant related work. First, we present theory-driven, rule-based approaches (Section 2.1), and then data-driven approaches (Section 2.2).

2.1 Theory-Driven Approaches

The seminal book on *Emotion and Meaning in Music* by Leonard Meyer [4] was one of the first to explicitly relate music expectation to principles of gestalt theory. This publication initiated major lines of research in music cognition and music theory. Tenney and Polansky [5] were among the first to define an implementable, quantitative model for detecting segment boundaries. Their model is based on the principles of proximity (in time) and similarity (in pitch). Several other models are based on gestalt principles as well: the Local Boundary Detection Model (LBDM) by Cambouropoulos [6, 7], the Grouper model by Temperley [8], the preference rules for grouping as defined in A Generative Theory of Tonal Music (GTTM) by Lerdahl and Jackendoff [9], the quantisation of these rules by Frankland and Cohen [10], the Implication-Realization theory by Narmour [11, 12], and the partial quantisation of this theory by Schellenberg [13]. More recently, vari-

ous theory-based approaches have been proposed by Rodríguez López [14]. A rule-based model not explicitly grounded on gestalt principles was proposed by Cenkerová et al. [15].

2.2 Data-Driven Approaches

Explicitly challenging the gestalt principles, Bod [16] introduces Data Oriented Parsing (DOP). A DOP-Markov parser learns probabilities for rewrite rules from a set of examples. One of the applications of this model was the prediction of segment boundaries in the Essen Folk Song Collection. In an error analysis, Bod shows that the DOP-Markov parser is able to learn regularities in phrase-ending patterns that do not adhere to gestalt rules.

Various data-driven approaches are based on information theory. Generally, a phrase boundary is inferred either before an unexpected melodic event or after an event for which the continuation is hard to predict. Methods differ in the way of computing the conditional probability of events given their preceding context. Juhasz [17] takes this approach to segment a collection of Hungarian folk songs. The multiple viewpoint statistical modelling method by Conklin and Witten [18] has been used for many symbolic music processing tasks such as generation, classification, and pattern discovery. The IDyOM model [19] employed the multiple viewpoint method for melodic segmentation. Lattner [20] employs a Restricted Bolzmann Machine to model the probability of a melodic event. This approach outperforms IDyOM, and sets the state-of-the-art for recognising phrase boundaries in the Essen Collection.

Rodríguez López [14] also introduced a data-driven component. A part of his segmentation system needs to be trained on a corpus.

3. DATA

An often used collection of segmented melodies is the Essen Folksong Collection (EFSC). This collection contains thousands of folk song melodies mainly from Germany, but also from other parts of Europe, and a relatively small number of melodies from other continents. In the process of creating this collection, the melodies have been segmented into phrases. Therefore, it offers a large amount of data on melodic segmentation which allows for statistical evaluation. Following earlier work, we use the database Erk, a subset of EFSC consisting of c. 1,700 melodies.

We also employ a recently published corpus from the Meertens Tune Collections (MTC), consisting of collections of thousands of instrumental and vocal songs from Dutch sources [21]. The collection we use in this paper is MTC-FS-INST-2.0, more specifically, those melodies that have lyrics, are dated after 1850, and have a time signature. This results in a selection of c. 7,500 melodies. For reasons that will be explained in section 3.1.2 we apply a further selection: from each tune family, we randomly select one melody. This results in a set of 1,323 melodies.

The third corpus we use is the collection of 371 harmonisations of chorales (CHOR) by Johann Sebastian Bach (1685–1750). [1] Since our focus is on melodic segmentation, we only use the melodies (i.e., the soprano parts).

Dataset	#songs	#boundary	#noboundary	total
MTC	1,323	7,054	63,856	70,910
ESSEN	1,632	7,703	62,490	70,193
CHOR	370	1,907	15,455	17,362

Table 1. Overview of the datasets indicating the number of songs and the sizes of the classes (number of 5-grams).

An overview of the datasets, the number of songs, and the class sizes is included in Table 1.

3.1 Some Caveats

Employing a collection of folk song melodies has some important consequences that have often not been discussed in previous work. We focus on two problems: tune family relations, and the rest as notational device.

3.1.1 Tune Families

One defining property of folk music is that it has been in oral circulation [22]. In the process of oral transmission, changes are introduced to the melodies and texts. Therefore, in a typical collection of folk songs, several variants of the same melody are included, exhibiting minor to large differences among each other. Such a group of related melodies is often designated as a *tune family* [23]. EFSC and MTC are no exceptions to this. For the collections in MTC, the tune families have largely been identified by collection specialists at the Meertens Institute. The tune family labels are included in the metadata that comes with the collection. For the ESFC this has not been done. From the titles of the songs, which are available in the metadata, it is clear that duplicates and variants of melodies are included, but there is no account of precisely which melodies are related.

The consequence of this for a data mining approach is that the independence of the train and test sets is not guaranteed since members of the same tune family may end up in both the train and test sets. Especially when the differences are small, this is problematic.

To solve this issue, we take advantage of the tune family labels as provided in the metadata of MTC.

3.1.2 Rests

In related work, the presence of a rest appeared to be a strong indicator of a phrase boundary. For example, one of the quantised GTTM preference rules (GPR 2a) states that the boundary strength is proportional to the length of a rest. In LBDM, next to pitch and inter-onset-intervals, rests are explicitly incorporated as one of the three features that contribute to the resulting local boundary strength. Furthermore, the occurrence of a rest is the first of Narmour's six conditions of melodic closure [11, p. 11].

There is, however, a difference between the meaning of a rest in composed music and in folk song transcriptions.

[1] This corpus is available as part of the humdrum-data repository: `https://github.com/humdrum-tools/humdrum-data`

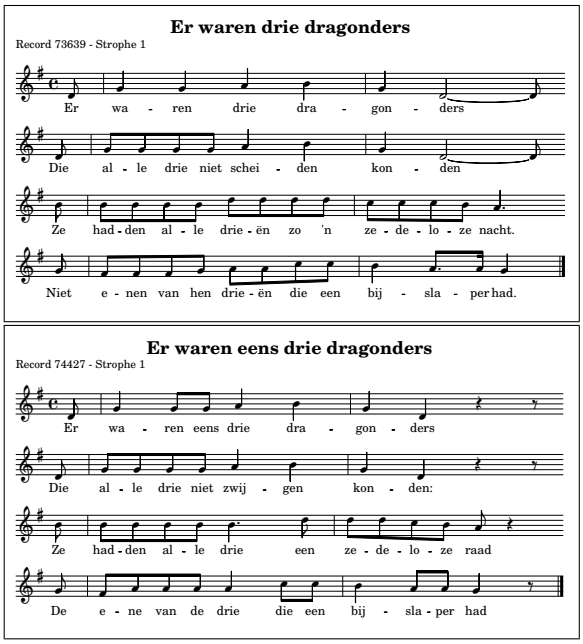

Figure 1. Transcriptions of two variants of the same melody showing different uses of the rest as notational device.

Again, this is related to the process of oral transmission. A composer typically uses common music notation as the primary device to communicate a piece of music to performers. Here, the notation of a rest is *prescriptive*, indicating the performers not to make sound. On the contrary, music notation as found in folk song collections is typically *descriptive*. The melodies have been transcribed from audio recordings, or from aural observation. Here, the rest is an indication of something a performer already has done. The example from the MTC that is shown in Figure 1 illustrates the resulting confusion that can arise if various transcribers contribute independently (or if one transcriber works inconsistently). In the upper transcription, the final note of each phrase is extended to fill the measure, while in the lower transcription, rests are included at the phrase boundaries. Crucially, inspecting the audio recordings that are the sources of these two transcriptions, [2] no noticeable differences are observable between the way the singers separate the phrases.

It appears that the rest as a symbol has a use in folk song transcription to represent a phrase boundary, rather than to indicate absence of sound. As a consequence, using the rest as a feature actually includes the ground truth in the feature set, which obviously results in an optimistic estimation of classification performance. We will therefore report results without using rests, and results including rests – and other ground truth dependent features – separately.

[2] These are available at `http://www.liederenbank.nl/index.php?lan=en` by entering the respective record numbers (73639 and 74427) in the search field.

4. METHOD

The approach in this paper largely is a feature engineering exercise. From previous studies and from general music theoretic considerations, we take inspiration of what features may contribute to the establishment of a phrase boundary. Next, we apply two machine learning algorithms: RIPPER and Random Forest. Thus, we do not take an a-priori theoretical basis, such as the gestalt principles, but we let the learning algorithm explore which features are of value and in what combination.

4.1 Objects and Features

The target of the classification is to find those notes after which a phrase ends. As object of classification we take each note in the melody with its local context of the two preceding and the two following notes, resulting in sequences of five notes, 5-grams. Since the aim is segmentation, the final phrase end, which also ends the melody, is excluded from the data set. Those 5-grams of which the third note is the final note of a phrase get the class label *boundary*, while all other 5-grams get the class label *noboundary*.

For each of the 5-grams we extract a large number of features. Each of those features can be considered a hypothesis of which information contributes to the concept of phrase boundary. We discern various groups of features. For extracting the feature values, the music21 toolkit has been used [24].

Elementary pitch features include for each of the five notes: the scale degree, the absolute pitch value in MIDI-representation, the interval with the previous note in semitones, the pitchcontour (up, down, equal), and the *Harmony* and the *Center of Gravity* as defined in [25].

Elementary rhythm features include the meter 'numerator' and 'denominator', the duration of the beat, the number of beats in the measure, and for each of the five notes: the metric weight, the duration (inter-onset-interval) in units of the beat-length, whether the note starts on or off the beat, and whether the duration increases for each of the first three notes. Furthermore, following the reasoning of Temperley [8, p. 70], we include a boolean feature that is True when the onset time of the fourth note is at the same position in the measure as the onset of the very first note of the melody. This accounts for the preference to start phrases at corresponding positions in the measures. To allow a more fine-grained version of this preference, we also include a boolean feature that is True if the onset time of the fourth note completes the time-span of a beat, starting the first time-span at the onset of the first note (which possibly is not on the beat in case of an anacrusis). The metric weight (beatstrength) and the length of the beat, both included as feature, are computed with the music21 toolkit.

Elementary lyric features (MTC only) include for each of the five notes: whether the lyric syllable is stressed, whether the lyric is a content word, whether the lyric syllable ends a content-word that rhymes with another content-word anywhere in the lyrics, whether the lyric syllable is the final syllable of a word, and whether the note is part

of a melisma. For labeling non-content words, detection of rhyme, and determining the word stress, the methods as described in [26] are used. Furthermore, we measure the distance between the third note of the 5-gram and the most recent rhyming syllable, both as number of notes and as number of beats.

Wherever applicable, we include the first-order contour of these elementary features as separate features, registering whether the value for a note is higher, equal, or lower than the value for the previous note. This provides the rule mining algorithm with relational information for the consecutive notes, which is beneficial because RIPPER is not able to include comparisons between features into the conditions that constitute the rules. Each condition consists of a single feature compared to an absolute value.

Next to these elementary features, we include features that are derived from previous models. For each of the five notes, the following values are included as feature:

- the sum of the values for the quantised GTTM GPRs 2a, 2b, 3a, and 3d, as defined in [10];
- the Local Boundary Strength as computed by the LBDM [7];
- the values for pitch proximity and pitch reversal as defined in [13];
- the prediction of Grouper [8];
- the information content as computed by the IDyOM model according to [19];
- features that are based on the conditions of closure as stated by Narmour [11, p. 11]: the metric weight contour for the third note, whether the third note is longer than the second, whether the interval between the first and second notes is larger than the interval between the second and third notes, and whether the direction of the melodic contour changes between the second and the third note.

Finally, we include several features that are not independent of earlier annotated segment boundaries. In an inspection of classification results, it appeared that often a boundary very close to the beginning of a phrase was predicted. To prevent this, we include the distance between the third note of the 5-gram and the beginning of the phrase, both as number of notes, and as number of beats. Furthermore, we include a boolean feature that is True if the onset of the fourth note is at the same position in the bar as the onset of the first note in the phrase. Lastly, we include for each of the five notes whether a rest follows the note.

In total, we have 162 features (excluding the class label), 31 of which are lyric features. [3] The lyric features are only computed for the MTC dataset, since lyrics are not present in the ESSEN and CHOR collections.

4.2 Learning Algorithms

RIPPER [27] is a rule mining algorithm that infers a set of classification rules from a data set. The basic procedure that is implemented in this algorithm is to split the training data into two folds (1/3 and 2/3), grow a rule on the 2/3 split, prune the rule using the 1/3 split, and remove the objects that are covered by the rule from the training set. This is repeated until no objects remain in the training set. Each iteration results in a rule that is added to the rule set. The algorithm starts finding rules that target the minority class, which is appropriate for the segmentation problem in which phrase boundaries are a minority class. The resulting rules are not independent. To reach a classification, the rules have to be applied in the order as provided by the mining algorithm. The advantage of a rule-set as resulting model is its interpretability. From the rules it is clear how a classification is established.

One important parameter of the RIPPER algorithm is the minimum number of objects per rule. By setting this to a low value, many rules result that might be too specific, while setting this to a high value results in less, and more general rules. We found that in general for our purpose 32 is a sensible value. Smaller values lead to much more rules, without considerably improving classification performance. Furthermore, since songs in general have much less than 32 phrase boundaries, this value forces the algorithm to generalise over songs. We use the implementation that is provided in the Weka workbench [28].

To better show the potential of the feature set, we also use a Random Forest classifier [29]. During training, a large number of decision trees are fitted to random subsets of the data. The classification is a majority vote of these individual trees. The models that result from this approach are not easily interpretable, but they generally reach a higher classification performance compared to a single decision tree or rule set. We experimentally found that the optimal number of trees in the forest is around 40. Larger forests do not considerably add to the classification performance. We use the implementation as provided in the Python module sklearn [30]. For evaluation, we employ a 5-fold cross-validation procedure, both for RIPPER and Random Forest. To further raise the independence between test and train sets in case of the Random Forest, we make the splits between the train and test sets at the level of melody. Thus, the 5-grams from the same melody all are either in the test or in the train set. For MTC, this implies that also tune families are always separated, while for EFSC and CHOR this cannot be guaranteed. The code for this paper is publicly available. [4]

5. RESULTS

Table 2 shows the classification results for the three datasets, the two classifiers, and various feature subsets.

5.1 General Remarks

The separate groups of elementary features (pitch, rhythm, lyrics) only reach moderate performance. Rhythm features consistently score better than pitch features. Lyric features clearly have considerable discriminative power,

[3] The full feature set is included in the supplementary material.

[4] https://github.com/pvankranenburg/ismir2020

	MTC					
	RIPPER			Random Forest		
Features	**Pr**	**Rc**	**F1**	**Pr**	**Rc**	**F1**
El. Pitch	0.58	0.17	0.26	0.43	0.26	0.32
El. Rhythm	0.75	0.53	0.62	0.72	0.57	0.63
El. Lyrics	0.64	0.38	0.48	0.56	0.43	0.49
El. NoLyr	0.73	0.61	0.67	0.80	0.58	0.68
El. All	0.77	0.73	0.75	0.85	0.69	0.76
Prev.	0.81	0.62	0.70	0.83	0.62	0.71
NoLyr	0.79	0.66	0.72	0.86	0.64	0.73
All	**0.82**	**0.76**	**0.79**	**0.89**	**0.72**	**0.80**
NoLyr+GT	0.84	0.80	0.82	0.90	0.76	0.82
All+GT	0.86	0.87	0.87	0.92	0.82	0.87

	EFSC					
	RIPPER			Random Forest		
Features	**Pr**	**Rc**	**F1**	**Pr**	**Rc**	**F1**
El. Pitch	0.57	0.18	0.27	0.49	0.31	0.38
El. Rhythm	0.78	0.53	0.63	0.77	0.62	0.69
El. Lyrics	-	-	-	-	-	-
El. NoLyr	0.78	0.63	0.69	0.83	0.69	0.76
El. All	-	-	-	-	-	-
Prev.	0.81	0.66	0.73	0.88	0.64	0.74
NoLyr	**0.83**	**0.68**	**0.75**	**0.90**	**0.70**	**0.79**
All	-	-	-	-	-	-
NoLyr+GT	0.90	0.88	0.89	0.95	0.87	0.90
All+GT	-	-	-	-	-	-

	CHOR					
	RIPPER			Random Forest		
Features	**Pr**	**Rc**	**F1**	**Pr**	**Rc**	**F1**
El. Pitch	0,68	0.49	0.57	0.77	0.65	0.71
El. Rhythm	0.76	0.66	0.71	0.84	0.69	0.76
El. Lyrics	-	-	-	-	-	-
El. NoLyr	**0.84**	**0.75**	**0.79**	**0.94**	**0.85**	**0.89**
El. All	-	-	-	-	-	-
Prev.	0.81	0.73	0.77	0.93	0.82	0.87
NoLyr	**0.85**	**0.77**	**0.81**	**0.95**	**0.86**	**0.90**
All	-	-	-	-	-	-
NoLyr+GT	0.94	0.84	0.89	0.98	0.91	0.94
All+GT	-	-	-	-	-	-

Table 2. Classification results (precision, recall, and F1 for the boundary class) on MTC, EFSC, and CHOR for various feature subsets, both for the rule miner (RIPPER) and for the Random Forest classifier. "El." denotes the elementary features. "NoLyr" denotes all features except for the lyrics features. "Prev." denotes the features from previous models. "GT" denotes the group of features that are not independent of the annotated phrase boundaries.

as is observable in the increase of the recall between the "El. NoLyr" and "El. All" subsets for MTC.

Comparing the performance between using elementary features only and using all features shows some improvement in the later case for MTC and EFSC, but not for CHOR. The "Prev." subset on its own consistently shows a good performance. This implies that the explainable power of the elementary features is comparable to the explainable power of the previous models. A large part of the boundaries remains unexplained with either which method.

Overall, MTC is the hardest to classify. Undoubtedly, this is a consequence of the careful compilation, ensuring only one melody per tune family. Since we have no tune family labels for EFSC and CHOR, the independence of train and test sets cannot be fully guaranteed. Therefore, the classification results might be too optimistic.

5.2 Rule Sets

The contents of the rules as found by the RIPPER algorithm reveals which features are paramount in detecting phrase boundaries. Although all rule sets give rise to interesting observations, it is not possible to discuss them all within the scope of this article. We show for two cases the first few rules, which typically cover many objects. As these rules are not directly derived from a theory of melodic perception, we are specifically interested to see to what extent the rules confirm existing understanding of melodic closure. Furthermore, these rules have the potential to lead to new hypotheses about what establishes closure in a melody.

First, we focus on the cases in which only elementary features are used. These are the first three discovered rules for MTC using the elementary pitch and rhythm features: [5]

```
Rule 0:
  (IOIbeatfractionthirdfourth = -) and
  (completesmeasuresong = True) and
  (IOIbeatfractionthird >= 1.25) and
  (meternumerator >= 4) and
  (IOIbeatfractionfirst <= 0.666667)
  => class=boundary (739.0/54.0)
Rule 1:
  (IOIbeatfractionthirdfourth = -) and
  (completesmeasuresong = True) and
  (IOIbeatfractionthird >= 1) and
  (IOIbeatfractionsecondthird = +) and
  (beatstrengthfourth >= 1)
  => class=boundary (705.0/88.0)
Rule 2:
  (IOIbeatfractionthirdfourth = -) and
  (completesmeasuresong = True) and
  (IOIbeatfractionthird >= 1.25) and
  (IOIbeatfractionfifth <= 1.5) and
  (VosHarmonyfourth >= 4) and
  (intervalsecond <= 0) and
  (diatonicpitchthird <= 30)
  => class=boundary (272.0/15.0)
```

Rule 0 classifies 739 5-grams from the train set correctly, and additionally covers 54 false positives. `IOIbeatfraction` denotes the duration of the note in units of the beat-length. The first rule mainly states that the fourth note should be shorter than the third, the third note ends at the position in the measure that is parallel to the start of the first note of the melody, the third note is longer than the beat (>=1.25 times), the first note is fairly short, and the `meternumerator` is 4. The last condition excludes all songs in e.g., 6/8 or 3/4 meter. The conditions that have been selected for these rules confirm considerations for several previous models. One of the central properties of a phrase-closing note seems to be its length, which should be longer than the beat. Furthermore, these rules all state that the length of the fourth note should be shorter than the third. This condition is present in 22 of the 30 discovered rules in this set. It contrasts with one of Narmour's conditions of closure [11, p. 11], which states that the closing note is longer than the previous note. Apparently, the data indicates that the relation with the next

[5] The full rule set is included in the supplemental material.

Dataset	RIPPER	Random Forest	IDyOM	Grouper	LBDM	Rest	Always
MTC	0.73 0.61 **0.67**	0.80 0.58 **0.68**	0.65 0.51 **0.57**	0.69 0.67 **0.68**	0.60 0.51 **0.55**	0.92 0.26 **0.40**	0.10 1.00 **0.18**
EFSC	0.78 0.63 **0.69**	0.83 0.69 **0.76**	0.71 0.49 **0.58**	0.70 0.61 **0.65**	0.60 0.47 **0.53**	0.96 0.31 **0.47**	0.11 1.00 **0.20**
CHOR	0.84 0.75 **0.79**	0.94 0.85 **0.89**	0.61 0.39 **0.47**	0.64 0.59 **0.62**	0.48 0.42 **0.45**	0.99 0.09 **0.17**	0.11 1.00 **0.20**

Table 3. Classification performance of related models. For each model, precision, recall, and F1 (bold) for the boundary class are reported. Results for RIPPER and Random Forest are for the featureset El. NoLyr.

note is more indicative, instead. One could speculate that the perception of closure at the third note is reinforced in retrospective when noticing that the next note is shorter. The rules that are found for the EFSC bear a similarity to those for MTC. The rules for CHOR differ more. But for all three data sets rhythm features dominate the top rules. This confirms earlier results as reported by Weyde [31]. The pitch features that are included mainly refer to pitch contour and the level of dissonance of the melodic interval (as registered by the features based on [25]).

Next, we consider the top rules that are discovered for MTC with the feature subset of all features ("All"): [6]

```
Rule 0:
  (grouperthird = True) and
  (rhymesthird = True) and
  (lbdmthird >= 0.280929)
  => class=boundary (2413.0/149.0)
Rule 1:
  (grouperthird = True) and
  (wordendthird = True) and
  (informationcontentfourth >= 7.252784) and
  (contourthird = -) and
  (lbdmfifth <= 0.159635)
  => class=boundary (641.0/33.0)
```

It is clear that the combined models of LBDM and Grouper, and the condition of rhyme constitute a very powerful rule that covers 2,413 boundary 5-grams in the training set, and only 149 noboundary 5-grams. In the second rule, also the information content as computed by IDyOM plays a role. But also some elementary features are used.

5.3 Existing Models

Table 3 shows a comparison with the performance of several existing models. The values for the RIPPER and Random Forest classifiers are those for the set of elementary features without the lyrics. This is not the best performing feature subset, but the larger subset would include IDyOM, Grouper and LBDM as features, which would not render a fair comparison. The IDyOM segmentation is computed with the implementation of IDyOM as available on GitHub. [7] Grouper is available as part of the Melisma Music Analyzer. [8] LBDM is implemented according to [7]. The threshold for peak-picking is chosen such that the resulting F1-value is maximised. The Rest model assumes a phrase boundary wherever a rest is notated in the score. This quantifies the effect of including the rest as a feature in a segmentation model that is evaluated on a collection of folk song melodies. As can be seen, the rest model typically results in high-precision, low-recall segmentation.

[6] The full rule set is included in the supplemental material.

[7] http://mtpearce.github.io/idyom/

[8] https://www.link.cs.cmu.edu/music-analysis/

Dataset	MTC	ESFC	CHOR
MTC	0.80 0.58 **0.68**	0.83 0.57 **0.67**	0.85 0.49 **0.62**
ESFC	0.76 0.61 **0.68**	0.83 0.69 **0.76**	0.83 0.68 **0.74**
CHOR	0.77 0.32 **0.45**	0.80 0.37 **0.51**	0.95 0.86 **0.90**

Table 4. Performance of cross-evaluation. The rows show the train sets, the columns the test sets. The values are: precision recall F1 (bold) for the boundary class.

For both MTC and EFSC the occurrence of a rest explains around 30% of the phrase boundaries. This is a considerable effect. Finally, a baseline model is included that classifies each note as a phrase boundary. The Random Forest classifier with the elementary feature set outperforms the other methods for ESFC and CHOR, and performs comparable to Grouper on MTC, although precision and recall are more balanced with Grouper. It also outperforms Lattner's RBM approach on EFSC (0.80 0.55 **0.63**) [20]. Notably the recall is higher. However, the currently presented approach is supervised and uses more features.

5.4 Cross Relations

We now examine the performance of the classifiers on the datasets they are not trained on. We use the Random Forest classifier and the feature subset "El. NoLyr". Most notable in the results as shown in Table 4 is the comparable cross-performance between ESFC and MTC. Apparently, the phrase endings in the Dutch and German folk song styles have comparable properties. The higher self-performance of ESFC might partly be caused by the tune-family problem (Section 3.1.1). The low performances of the classifiers trained on CHOR are mainly caused by low recall. This could indicate that some types of phrase endings that occur in MTC and ESFC are absent in CHOR.

6. CONCLUSION

We presented an approach to melodic segmentation that builds on and integrates elementary melodic features and existing segmentation models in a theory-agnostic way. By deriving a rule-set using a large number of features, we get an indication of which features are crucial for detecting phrase boundaries in melodies. A notable observation is that a phrase boundary is mainly detectable with rhythm features. By employing a Random Forest classification, we get an indication of the discriminative power of the considered feature sets. The resulting classifier outperforms all earlier approaches to the problem of automatic melodic segmentation. By cross-evaluation, we detect a connection between MTC and EFSC.

7. REFERENCES

[1] D. Conklin, "Melodic analysis with segment classes," *Machine Learning*, 2006.

[2] H. Schaffrath, Ed., *The Essen Folksong Collection*. Stanford, CA: Center for Computer Assisted Research in the Humanities, 1995.

[3] K. Koffka, *Principles of Gestalt psychology*. Harcourt, Brace and Company, 1935.

[4] L. B. Meyer, *Emotion and Meaning in Music*. Chicago: University of Chicago Press, 1956.

[5] J. Tenney and L. Polansky, "Temporal gestalt perception in music," *Journal of Music Theory*, vol. 24, no. 2, pp. 205–241, 1980.

[6] E. Cambouropoulos, "Musical rhythm: A formal model for determining local boundaries, accents and metre in a melodic surface," in *Music, Gestalt, and Computing: Studies in Cognitive and Systematic Musicology*, M. Leman, Ed. Springer Verlag, 1997, pp. 277–293.

[7] ——, "The local boundary detection model (lbdm) and its application in the study of expressive timing," in *Proc. of the Intl. Computer Music Conf*, 2001.

[8] D. Temperley, *The Cognition of Basic Musical Structures*. Cambridge, MA: MIT Press, 2001.

[9] F. Lerdahl and R. Jackendoff, *A Generative Theory of Tonal Music*. Cambridge, Mass.: MIT Press, 1983.

[10] B. W. Frankland and A. J. Cohen, "Parsing of melody: Quantification and testing of the local grouping rules of lerdahl and jackendoff's a generative theory of tonal music," *Music Perception*, vol. 21, no. 4, pp. 499–543, 2004.

[11] E. Narmour, *The Analysis and Cognition of Basic Melodic Structures - The Implication-Realization Model*. Chicago and Londen: The University of Chicago Press, 1990.

[12] ——, *The Analysis and Cognition of Melodic Complexity: the Implication-Realization model*. Chicago: University of Chicago Press, 1992.

[13] E. G. Schellenberg, "Expectancy in melody: tests of the implication-realization model," *Cognition*, vol. 58, no. 1, pp. 75–125, 1996.

[14] M. E. Rodríguez-Lopez, "Automatic melody segmentation," Ph.D. dissertation, Utrecht University, Utrecht, 2016.

[15] Z. Cenkerová, M. Hartmann, and P. Toiviainen, "Crossing phrase boundaries in music," in *Proceedings of the 15th Sound and Music Computing Conference 2018*, 2018, pp. 66–71.

[16] R. Bod, "Memory-based models of melodic analysis: Challenging the gestalt principles," *Journal of New Music Research*, vol. 31, no. 1, pp. 27–37, 2002.

[17] Z. Juhász, "Segmentation of hungarian folk songs using an entropy-based learning system," *Journal of New Music Research*, vol. 33, no. 1, pp. 5–15, 2004. [Online]. Available: http://dx.doi.org/10.1076/jnmr.33.1.5.35395

[18] D. Conklin and I. H. Witten, "Multiple viewpoint systems for music prediction," *Journal of New Music Research*, vol. 24, no. 1, pp. 51–73, 1995.

[19] M. T. Pearce, D. Müllensiefen, and G. A. Wiggins, "Melodic grouping in music information retrieval: New methods and applications," in *Advances in Music Information Retrieval*, ser. Studies in Computational Intelligence, Z. W. Raś and A. A. Wieczorkowska, Eds. Berlin, Heidelberg: Springer, 2010, vol. 274, pp. 364–388.

[20] S. Lattner, C. E. Cancino Chacón, and M. Grachten, "Pseudo-supervised training improves unsupervised melody segmentation," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, Q. Yang and M. J. Wooldridge, Eds., 2015, pp. 2459–2465.

[21] P. Van Kranenburg and M. De Bruin, "The meertens tune collections: Mtc-fs-inst 2.0," Meertens Institute, Amsterdam, Meertens Online Reports 2019-1, 2019.

[22] R. Elbourne, "The question of definition," *Yearbook of the International Folk Music Council*, vol. 7, pp. 9–29, 1975.

[23] S. Bayard, "Prolegomena to a study of the principal melodic families of british-american folk song," *Journal of American Folklore*, vol. 63, no. 247, pp. 1–44, 1950.

[24] M. S. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data," in *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)*, 2010, pp. 637–642.

[25] P. G. Vos and D. Pasveer, "Goodness ratings of melodic openings and closures," *Perception & Psychophysics*, vol. 64, no. 4, pp. 631–639, 2002.

[26] P. Van Kranenburg and F. Karsdorp, "Cadence detection in western traditional stanzaic songs using melodic and textual features," in *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, 2014, pp. 391–396.

[27] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the Twelfth International Conference on Machine Learning*, 1995, pp. 115–123.

[28] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques"*, fourth edition ed. Morgan Kaufmann, 2016.

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[31] T. Weyde, "On the influence of pitch on melodic segmentation," in *Proceedings of the Fifth International Conference on Music Information Re- trieval*, Barcelona, 2004.