Rule Mining for Local Boundary Detection in Melodies

Peter van Kranenburg - Meertens Instituut, Universiteit Utrecht

Approach

1. Extract 5-grams of Notes
2. Extract Feature Values
3. Perform Rule Mining (RIPPER)
4. Interpret Results

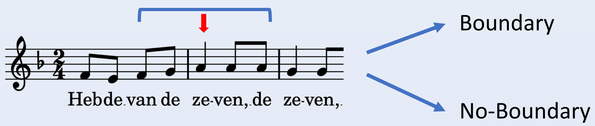
Data sets

1. Meertens Tune Collections (MTC)
2. Essen Folk Song Collection (EFSC)
3. Bach Chorale Melodies (CHOR)

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

Sizes of the data sets.

Extract 5-grams



Extract Feature Values

1. Elementary Pitch (58) and Rhythm (30)
2. Lyrics (30)
3. Existing models (51)
4. Ground-truth dependent (9)

Rule Mining

MTC (El. Pitch and Rhythm)

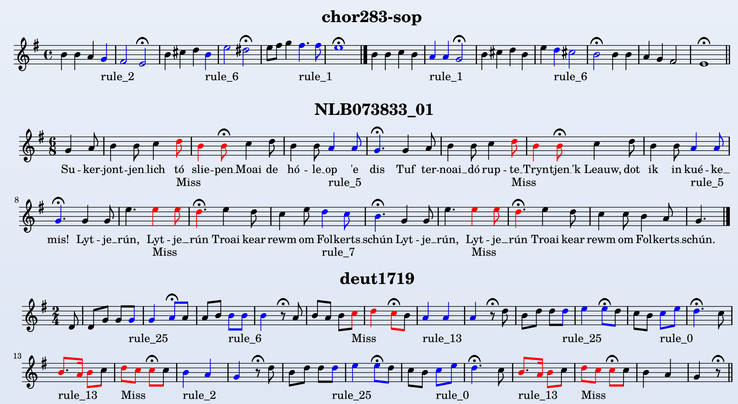
```
Rule 0:
  (IOIbeatfractionthirdfourth = -) and
  (completesmeasuresong = True) and
  (IOIbeatfractionthird >= 1.25) and
  (meternumerator >= 4) and
  (IOIbeatfractionfirst <= 0.666667)
  => class=boundary (739.0/54.0)
Rule 1:
  (IOIbeatfractionthirdfourth = -) and
  (completesmeasuresong = True) 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 3:
  (IOIbeatfractionthirdfourth = -) and
  (completesmeasuresong = True) and
  (beatstrengthfirst <= 0.5) and
  (IOIbeatfractionthird >= 1.333333) and
  (meternumerator >= 4) and
  (beatstrengthsecond <= 0.25)
  => class=boundary (136.0/14.0)
Rule 4:
  (IOIbeatfractionthirdfourth = -) and
  (completesmeasuresong = True) and
  (VosHarmonyfourth >= 4) and
  (intervalfifth >= 0) and
  (IOIbeatfractionfifth <= 0.333333) and
  (midipitchfourth <= 67) and
  (beatduration >= 1.5)
  => class=boundary (102.0/12.0)
Rule 5:
  (IOIbeatfractionthirdfourth = -) and
  (completesmeasuresong = True) and
  (intervaldirthirdfourth = +) and
  (intervalfifth >= 0) and
  (diatonicpitchthirdfourth = =)
  => class=boundary (436.0/92.0)
```

EFSC (El. Pitch and Rhythm)

```
Rule 0:
  (completesmeasuresong = True) and
  (IOIbeatfractionthirdfourth = -) and
  (contourthird = -) and
  (beatstrengthfirst <= 0.5) and
  (IOIbeatfractionthird >= 1.333333) and
  (IOIbeatfractionfifth <= 1)
  => class=boundary (1284.0/41.0)
Rule 1:
  (completesmeasuresong = True) and
  (IOIbeatfractionthirdfourth = -) and
  (contourthird = -) and
  (intervalfifth >= 0) and
  (scaledegreefourth <= 5) and
  (meternumerator >= 4) and
  (beatstrengthfifth >= 0.25)
  => class=boundary (316.0/32.0)
Rule 2:
  (completesmeasuresong = True) and
  (IOIbeatfractionsecondthird = +) and
  (IOIbeatfractionfourthfifth = =) and
  (IOIbeatfractionthird >= 1.5) and
  (contoursecond = -)
  => class=boundary (446.0/44.0)
Rule 3:
  (completesmeasuresong = True) and
  (IOIbeatfractionthirdfourth = -) and
  (contourthird = -) and
  (intervaldirthirdfourth = +) and
  (intervalfifth >= 0) and
  (beatcount <= 2) and
  (VosCenterGravitysecond = True)
  => class=boundary (100.0/7.0)
Rule 4:
  (completesmeasuresong = True) and
  (IOIbeatfractionsecondthird = +) and
  (IOIbeatfractionfifth <= 1) and
  (IOIbeatfractionthird >= 1.333333) and
  (beatstrengthfourthfifth = -)
  => class=boundary (318.0/36.0)
Rule 25:
  (completesmeasuresong = True) and
  (beatstrengththird >= 0.5) and
  (IOIbeatfractionfifth <= 0.75) and
  (beatcount <= 2) and
  (beatstrengthfirst <= 0.25) and
  (beatstrengthfourthfifth = +)
  => class=boundary (170.0/54.0)
```

CHOR (El. Pitch and Rhythm)

```
Rule 0:
  (completesmeasuresong = True) and
  (intervaldirthirdfourth = +) and
  (IOIbeatfractionfourthfifth = =) and
  (IOIbeatfractionfirstsecond = +) and
  (beatstrengththirdfourth = -)
  => class=boundary (257.0/10.0)
Rule 1:
  (completesmeasuresong = True) and
  (IOIbeatfractionthird >= 2) and
  (IOIbeatfractionfourthfifth = =) and
  (beatstrengthsecond <= 0.25)
  => class=boundary (368.0/4.0)
Rule 2:
  (completesmeasuresong = True) and
  (VosHarmonyfourth >= 4) and
  (contourfourth = +) and
  (contourfirst = -)
  => class=boundary (255.0/42.0)
Rule 3:
  (completesmeasuresong = True) and
  (VosHarmonyfourth <= 0) and
  (IOIbeatfractionfifth <= 1) and
  (beatstrengthfourth <= 0.25)
  => class=boundary (320.0/79.0)
Rule 4:
  (completesmeasuresong = True) and
  (VosHarmonyfourth >= 4) and
  (IOIbeatfractionfirstsecond = +) and
  (diatonicpitchthird <= 30) and
  (contoursecond = -)
  => class=boundary (47.0/3.0)
Rule 5:
  (completesmeasuresong = True) and
  (intervalfourth >= 0) and
  (beatstrengththird >= 0.5) and
  (IOIbeatfractionfourthfifth = =) and
  (VosHarmonyfifth <= 0) and
  (VosCenterGravityfourth = False)
  => class=boundary (40.0/3.0)
Rule 6:
  (IOIbeatfractionthird >= 2) and
  (intervaldirthirdfourth = +) and
  (beatstrengthfirstsecond = +) and
  (beatcount >= 4)
  => class=boundary (137.0/5.0)
```



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**

Comparison with existing segmentation algorithms. Values are Precision, Recall, and F1

	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	-	-	-	-	-	-

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.