Composer classification using grammatical inference

Jeroen Geertzen Menno van Zaanen

ILK
Dept. of Communication & Information Sciences
Tilburg University
Tilburg, The Netherlands

{j.geertzen, mvzaanen}@uvt.nl

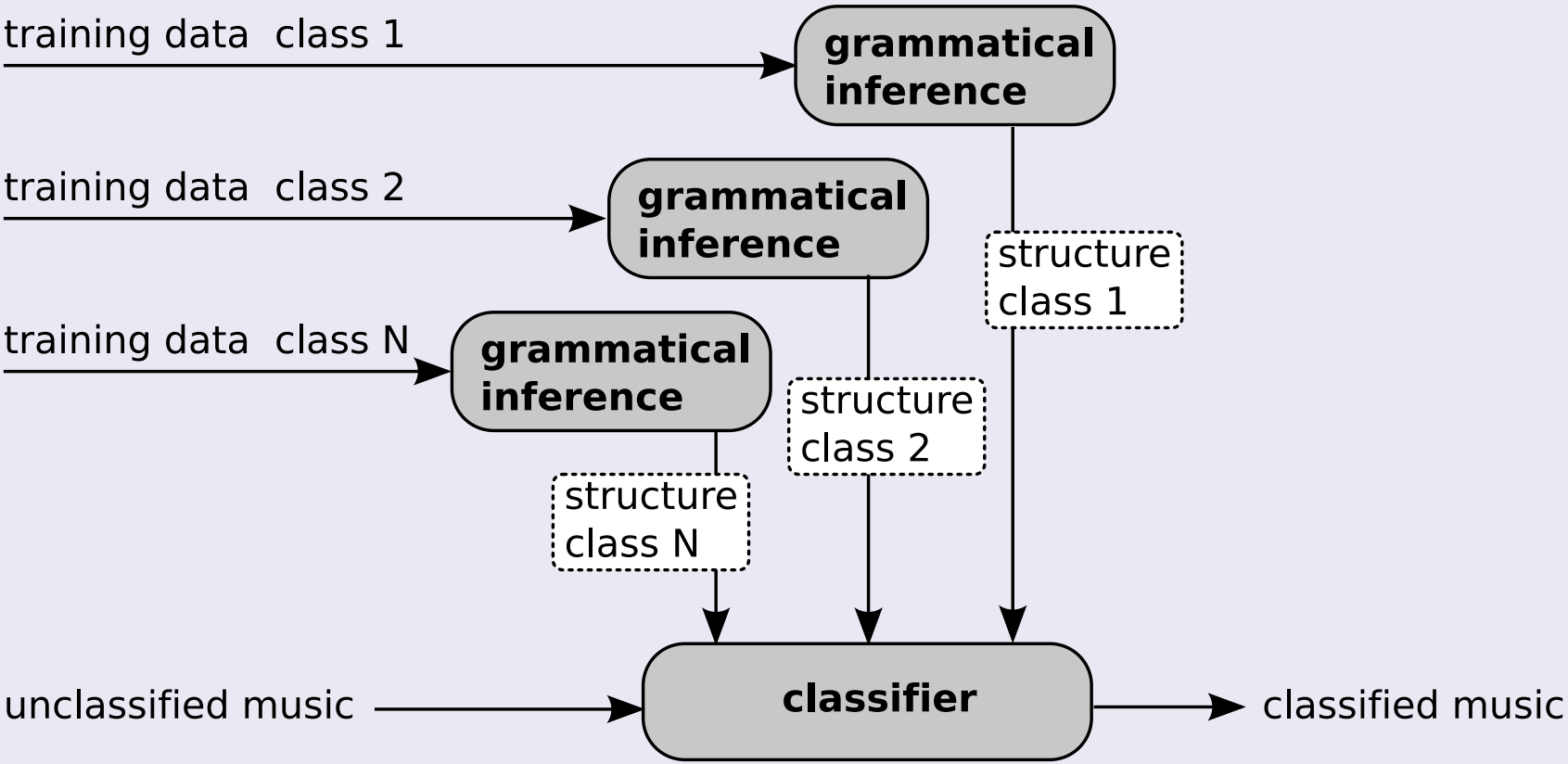
Overview

- Introduction
- Grammatical inference
- Regular expression-based classifier
- Results
- Conclusion and future work

Introduction

- Task:
  - Composer classification
    - Given a musical piece
    - Decide who composed it
    - (Given a predetermined set of composers)
  - Similar to genre classification
- Approach:
  - Find significant patterns in music
    Grammatical inference
  - Use patterns to identify composer
    Regular expression-based classifier

Approach


training data class 1
grammatical inference
training data class 2
grammatical inference
structure class 1
training data class N
grammatical inference
structure class 2
structure class N
unclassified music
classifier
classified music

Grammatical inference

- Automatically learn structure from plain sequences
- Alignment-Based Learning (ABL)
  - Developed to work on natural language
  - Based on substitutability test

Alignment-Based Learning

1. Alignment learning
2. Selection learning

Alignment learning

Test for constituents (Harris, 1951):

Elements of the same type are substitutable

What is $(_{NP}$ *a family fare*$)_{NP}$

Replace noun phrase with another noun phrase

Alignment learning

Test for constituents (Harris, 1951):

Elements of the same type are substitutable

What is $(_{NP}$ *a family fare*$)_{NP}$

Replace noun phrase with another noun phrase

What is $(_{NP}$ *the payload of an African Swallow*$)_{NP}$

Alignment learning

Test for constituents (Harris, 1951):

Elements of the same type are substitutable

What is $(_{NP}$ *a family fare*$)_{NP}$

Replace noun phrase with another noun phrase

What is $(_{NP}$ *the payload of an African Swallow*$)_{NP}$

Reverse:

What is	*a family fare*
What is	*the payload of an African Swallow*

Alignment learning

Test for constituents (Harris, 1951):

Elements of the same type are substitutable

What is $(_{NP}$ *a family fare*$)_{NP}$

Replace noun phrase with another noun phrase

What is $(_{NP}$ *the payload of an African Swallow*$)_{NP}$

Reverse:

What is $(_{X}$ *a family fare*$)_{X}$

What is $(_{X}$ *the payload of an African Swallow*$)_{X}$

- Align pairs of sentences
- Unequal parts of sentences are stored as hypotheses
- Align all sentences against all other sentences

Selection learning

- Alignment learning can generate overlapping brackets

$$\text{from } (_{Y_1} \text{ Tilburg } (_{X_2} \text{ to})_{Y_1} \text{ Helsinki})_{X_2}$$
$$\text{from } (_{X_1} \text{ Helsinki } (_{Y_2} \text{ to})_{X_1} \text{ Tilburg})_{Y_2}$$

- Underlying grammar is considered context-free
- Structure describes parse according to underlying grammar
- "Wrong" brackets have to be removed

From structure to context-free grammars to regular expressions

1. Alignment-Based Learning finds structure

Example

$(_Z$ a $(_Y$ b $(_X$ c$)_X$ d$)_Y)_Z$

From structure to context-free grammars to regular expressions

1. Alignment-Based Learning finds structure
2. Structure can be converted into context-free grammar rules

Example

$(_Z$ a $(_Y$ b $(_X$ c$)_X$ d$)_Y)_Z$

$$
\begin{aligned}
Z &\rightarrow a\ Y \\
Y &\rightarrow b\ X\ d \\
X &\rightarrow c
\end{aligned}
$$

From structure to context-free grammars to regular expressions

1. Alignment-Based Learning finds structure
2. Structure can be converted into context-free grammar rules
3. CFG rules can be converted into regular expressions

Example

$(_Z$ a $(_Y$ b $(_X$ c $)_X$ d $)_Y)_Z$

$$
\begin{aligned}
Z &\rightarrow \text{a } Y \\
Y &\rightarrow \text{b } X \text{ d} \\
X &\rightarrow \text{c}
\end{aligned}
$$

```
a .*
b .* d
c
```

Regular expression-based classifier

Input
- Set of ⟨regular expression, class⟩ pairs
- Piece of music

Output
- Class (composer)

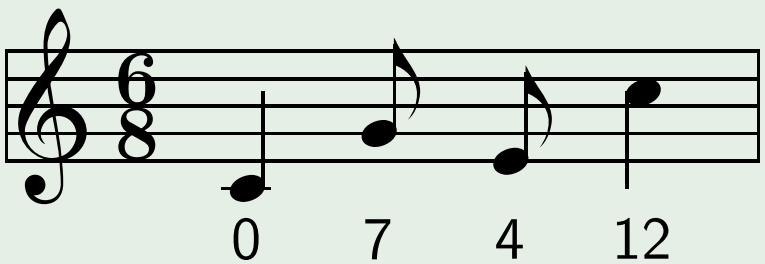
Internals

1. Take piece of music to be classified
2. Apply all regular expressions
3. Return class that has most regular expressions matching

Encoding pitch

- pitch absolute

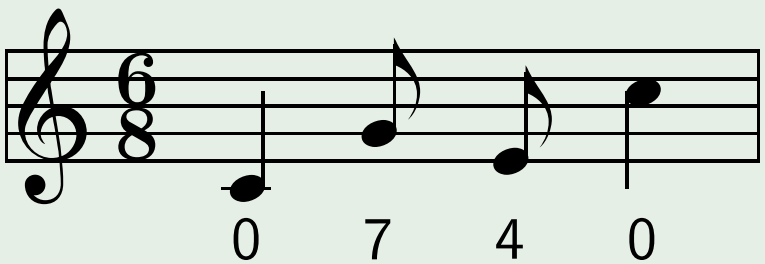
Example



Encoding pitch

- pitch absolute
- pitch absolute modulo octave

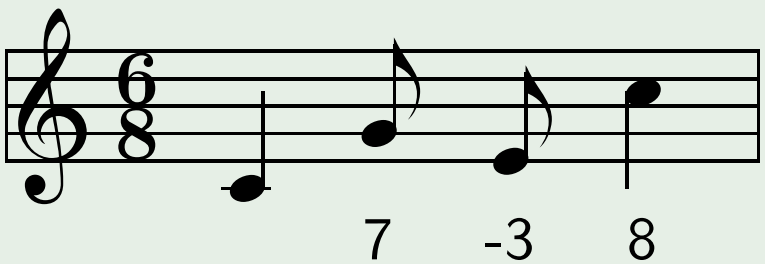
Example



Encoding pitch

- pitch absolute
- pitch absolute modulo octave
- pitch relative (to previous note)

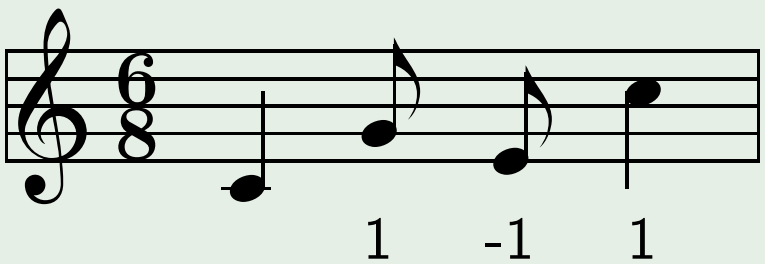
Example



Encoding pitch

- pitch absolute
- pitch absolute modulo octave
- pitch relative (to previous note)
- pitch contour (to previous note)

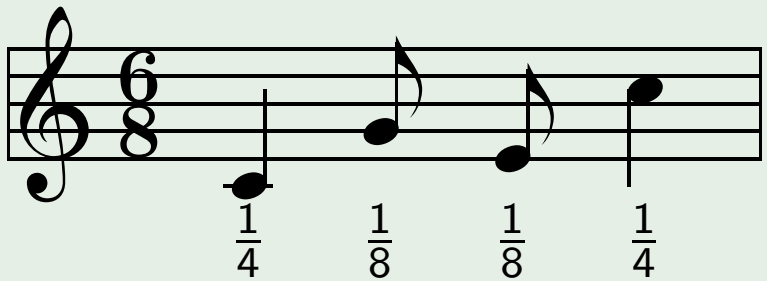
Example



Encoding duration

- duration absolute

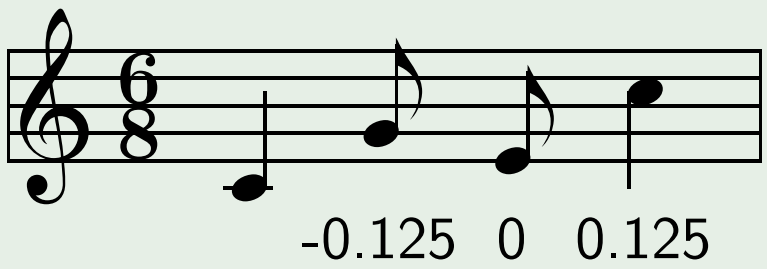
Example



Encoding duration

- duration absolute
- duration relative (prev note) subtraction

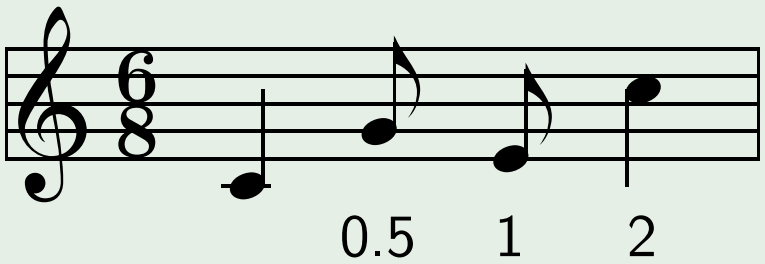
Example



Encoding duration

- duration absolute
- duration relative (prev note) division

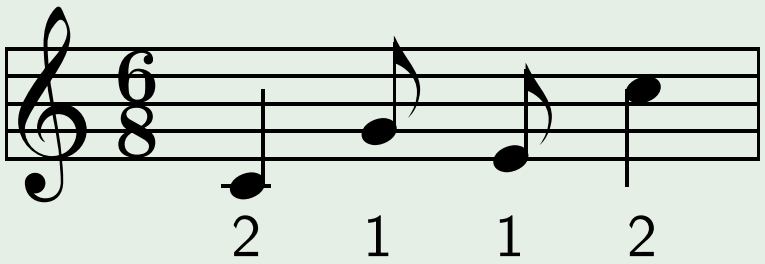
Example



Encoding duration

- duration absolute
- duration relative (prev note)
- duration absolute relative to meter

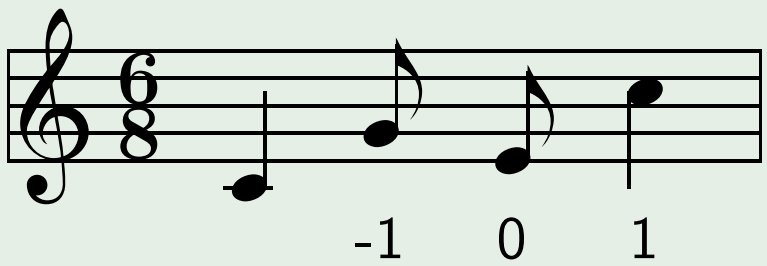
Example



Encoding duration

- duration absolute
- duration relative (prev note)
- duration absolute relative to meter
- duration relative (prev note) relative (meter) subtraction

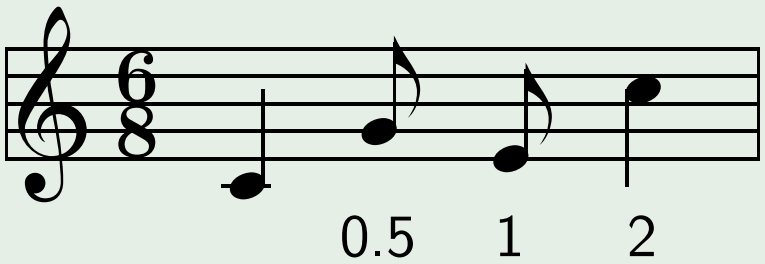
Example



Encoding duration

- duration absolute
- duration relative (prev note)
- duration absolute relative to meter
- duration relative (prev note) relative (meter) division

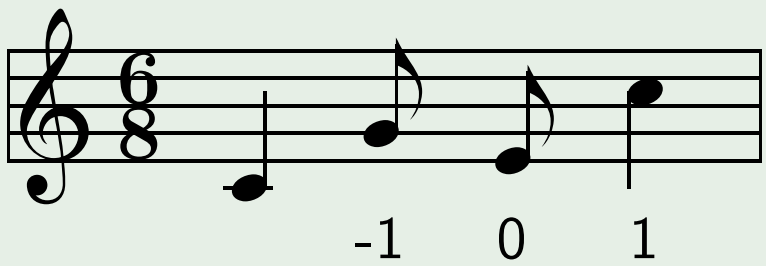
Example



Encoding duration

- duration absolute
- duration relative (prev note)
- duration absolute relative to meter
- duration relative (prev note) relative (meter)
- duration contour (to previous note)

Example



Data set

- Humdrum **kern format (symbolic)
- Use first voice only

Baroque Preludes: Bach (42), Chopin (24)

Classic Quartet: Beethoven (70), Haydn (213), Mozart (82)

Settings

- 2^{nd} order Markov model (MM-2) as baseline
- Pitch relative, duration relative encodings
- Leave-one-out cross-validation (small amount of training data)
- Error rate: $\frac{\#\text{ incorrect pieces}}{\#\text{ pieces}}$

Results (error rates)

Dataset	*Dimension*	*ABL*	*MM-2*
BAROQUE	MELODY	19.8 ±0.3	29.1 ±0.4
	RHYTHM	22.5 ±0.6	32.4 ±0.7
	JOINT	19.9 ±0.2	29.0 ±3.6
CLASSIC	MELODY	23.6 ±0.7	34.8 ±2.1
	RHYTHM	28.8 ±1.2	37.2 ±5.9
	JOINT	21.3 ±1.3	35.1 ±2.8

Conclusion

- Grammatical inference approach to composer classification
- Finds global patterns (of arbitrary length)
- Patterns are used in regular expression classifier
- Outperforms Markov models

Future work

- Try different (combinations of) encodings of music
- Expand data collection (more pieces; more composers)
- Use regular expressions as features in standard ML classifier
- Try other ways of converting grammar into regular expressions
- Use probabilities from learned probabilistic grammar

Previous work

- E. Backer and P. van Kranenburg, “On musical stylometry—a pattern recognition approach”, in Pattern Recognition Letters, 26 (2005), 299–309.
- E. Pollastri, G. Simoncelli, “Classification of Melodies by Composer with Hidden Markov Models”, wedelmusic, p. 0088, First International Conference on WEB Delivering of Music (WEDELMUSIC’01), 2001.