Melodic Feature Machines
MELFEATURE & MELEX

Klaus Frieler
University of Hamburg

Symposium on MIR
Utrecht February, 20th 2007

Introduction

- We propose a general purpose system for description and generation of melodic features and transformations.
- Based on the general theory of sequence transformations with special consideration of rhythm & melody transformations.
- Related Work: Conklin‘s (Multiple) Viewpoints, Cambouropoulos‘ Parametric Profiles, Hurons Humdrum, M2K, DIY…

Features & Melodic Analysis

- Features are the main building blocks for (automated) description and analysis of melodies.
- Features add or remove information from a melody.
- Features should reflect (in some way) moments of cognitive processing.

The Transformational Approach

- The main characteristic of music is its unfolding in time, order of events are crucial

⇨ Sequences, generalised time-series

- Analysis of objects is defined by admissible and typical sets of operations

⇨Notion of sequence transformation

Sequences, Rhythm & Melodies

- Let **X** be a set. A sequence of length |s| = N over X is a map

$$s : [\,0\,,\,N\text{-}1\,] \rightarrow \mathbf{X}$$

- Notation: Seq(**X**) **(**Sequences over **X**)
- A rhythm is a sequence with $\rho(k) = (t_k, e_{k,} d_{k,,})$, where $t_k < t_{k+1}$, e_k are events, and d_k duration
- A melody is a rhythm with events p_k in a tone system **T**, e.g. $\mu(k) = (t_k, p_k , d_{k,})$,

Sequence Transfomations

- A (unary) sequence transformation T is a mapping of sequences over a set **X** to sequences over a set **Y**.

$$T: \text{Seq}(\mathbf{X}) \rightarrow \text{Seq}(\mathbf{Y})$$

- Generalisation to n-ary sequence transformation obvious.
- Example for a binary sequence transformation: Melodic similarity

Sequence Transformations

- A sequence transformation can be:
  - contracting: |Ts|<|s|
  - point-wise: |Ts| = |s|
  - expanding |Ts|> |s|
  - a (global) feature :|Ts| = const.
  - None of the above
- Sequence transformation all called compound if

$$Y = Y_1 \times Y_2 \times \ldots Y_n$$

Melody and Rhythm Transformations

- Examples of pitch transformations:
  - Pitch intervals (PI) (reducing)
  - Pitch projection (P), Pitch classes (PC) (pointwise)
  - Contour /Parson‘s Code (pointwise)
- Examples of rhythmic transformations:
  - IOI‘s/Durations (D) (reducing)
  - Duration classes (reducing)
  - Metric annotation (pointwise)

Example: n-gram Transformation

- n-grams are subsequences of length n.
- n-gram transformation:

  n: Seq(**X**) ⇨ Seq(Seq(**X**))
- Example: 2-grams

  n(s)(0) = { s(0), s(1) }

  n(s)(1) = { s(1), s(2) }

  n(s)(2) = { s(2), s(3) }

  …

Sequence Transformations

- Many transformations can be done for all sequences, e.g. n-grams, descriptive statistics
- Many transformation can be constructed as chains and products of other transformations
- Many transformations depend only on the *class* of the sets X, Y, e.g. differentiation, translations, projections, similarity computations, clustering.

Melfeature – A feature machine

- *Melfeature* is a cross-platform console application
- Two basic modes: Feature and transformation mode
- Takes csv-formatted representation of melodies (MCSV, as *Simile*) and gives CSV, R, and WEKA output.
- Successfully used already for investigations of melodic accents (cf. Müllensiefen‘s talk today) and n-gram complexity (cf. my talk at ICMPC9 in Bologna)

Melfeature - Status Quo

- Implemented Transformations:
  - Over 40 single accents rules (can be partly used as markers, e.g. syncopations on various metrical levels)
  - Optimised joint accent rules
  - Jensen-Shannon-Divergence

Melfeature- Status Quo

- Implemented Features:
  - Descriptive statistics of pitch, pitch intervals, (quantized) durations sequences: Range, Minimum, Maximum, Average, Median, Modus, Variance, Skew, Kurtosis, Entropy
  - Huron-Contour (simple, standard)
  - N-gram complexitiy (Dynamic, N=1…4, each on P, PI, D)
  - N-gram Zipf-complexity (N=1…4, each on P, PI, D)

Melfeature – Quo Vadis?

- Limitation: Not truly modular (every feature got a name and flag).
- A scripting mechanism which allows arbitrary combinations of transformations is strongly needed.
- Integration in or as base of a more general melodic analysis framework (e.g. *Simile*, M^4S, *Melex…)*

Melex – A melody explorer

- Web-based exploration of folk songs using the transformational approach.
- Written entirely in Javascript.
- Melodies are contained in HMTL-pages using EsAC
- Every song can be listened to in a dead-pan MIDI rendition.
- URL: http://melex.mu-on.org

Melex – A melody explorer

Purposes (so far):

- Education
- Demonstration, proof of transformational concept
- Prototype for a future system
- Generating of hypothesises‘
- Just to play around…

Melex – A melody explorer

- User can choose between analysis of single folksong melodies and (subsets of) entire collections
- User can choose basic transformation:
  - Normalized pitch, pitch class, diatonic pitch, intervals, Parsons Code, durations, duration ratios
- User can choose main transformation: 1-4-grams
- User can choose output between descriptive statistics, simple histograms or just displaying transformations.

Melex – Quo Vadis?

- Vision:
  - Expand and integrate *Melfeature* and possibly *Simile* to a server-side application,
  - use a melody database,
  - add fancy notation features, database upload, and result export, and
  - get a full-fledged, web-based melody analysis workbench called *Melex*...

Conclusion & Outlook

- The modular, transformational approach has been shown and is believed to be a powerful approach to melodic analysis.
- In fact: Only systematises and generalises what has been doing in computational melody analysis so far, and maybe what can be done at all.

Melodic Feature Machines

MELFEATURE & MELEX

Thank you!