drizo research

oct.08-may.09
Objective

- Improve similarity computation with trees
- Focus on melodic reduction and tree distances
- Monodies
- Variation form
- Analised corpus
  - k331 and k265 plus variations
  - Referenced from early works on comparison such as Sankoff, 90
Other methods

- Study of other tree representations that consider melodic reduction (see next slide)
  - GTTM: Hamanaka implementation
  - Schenkerian analysis: Marsden
  - Smoliar: grammars
- None contain a simple propagation method ready to be used
  - Available implementations: not usable with real data
  - My point of view: problem cannot be modeled by rules
- Conclusion: try to learn
Fig. 5. Schenker’s analysis of the first phrase of Mozart’s piano sonata in A, K.331 (Schenker, 1935, Fig. 157).
Learning

- Musical Theory: after a correct reduction two variations of the same source theme should yield this source theme
- Tag manually several variations of the same songs
  - with melodic and tonal analysis
  - with a possible reduction
    - Only two songs so far
    - But ....
    - Evidences that show that something can be learnt
k331

Very different trees

Theme

Var. 1
k331 manual reductions

Theme
First reduction of theme

Var1
First reduction of var1

Last reduction of both
Trees fully tagged

Using graphical tool developed to introduce manually trees

However, highlighted nodes and leaves are the only ground truth, inner nodes are unknown
Line of work

- In DRIMS project
- Given a corpus of tagged reductions learn automatically the way to propagate
  - evaluation method: similarity between variations
  - this is what Marsden method lacks in his huge work
- Possible methods: probabilistic, ILP
And now?

- Research line stalled because of the corpus problem
- We wanted a better method to compare trees
- If not possible, why don’t we compare partially tagged trees?
New tree edit distance

- Designed for our problem

**Rationale:**
- recursive top-down distance
- change substitution cost between two nodes used in other distance

(next slide)
Both labels are non-empty

\[ \text{cost}_{\text{sub}}(T_1, T_2) = \text{cost}_{\text{label-sub}}('x', 'y') \]
One label is non-empty

cost_{subst}(T_1, T_2) =
1/2 * (cost_{labelsubst}(‘x’,‘y’) + cost_{labelsubst}(‘x’,‘z’))

Average of possible costs. Recursive downwards

NOTE: maybe the cost of deleting ‘y’ or ‘z’ should be added
Both labels are empty

Distance of forests \{w,x\} vs. \{y,z\}
See as the edit distance between sequences wx and yz where each symbol is a tree
Distance

Let $a = l(a_1, ..., a_T)$ and $b = l'(b_1, ..., b_V)$ be two trees. The distance between two trees $d(a : b)$ and the distance $d'(a_1, ..., a_T : b_1, ..., b_V)$ can be defined as:

$$d(a : b) = \begin{cases} 
  c_{sust}(l, l'), \text{if leaf}(a) \land \text{leaf}(b) \\
  \frac{\sum_{j=1}^{V} d(a : b_j)}{V}, \text{if leaf}(a) \land \neg \text{leaf}(b) \\
  \frac{\sum_{i=1}^{T} d(a_i : b)}{T}, \text{if } \neg \text{leaf}(a) \land \text{leaf}(b) \\
  d'(a_1, ..., a_T : b_1, ..., b_V), \text{otherwise}
\end{cases}$$

$$d'(a_1, ..., a_T : b_1, ..., b_V) = \min \left\{ 
  d'(a_1, ..., a_{T-1} : b_1, ..., b_V) + c_{del}(a_T), \\
  d'(a_1, ..., a_T : b_1, ..., b_{V-1}) + c_{ins}(b_V), \\
  d'(a_1, ..., a_{T-1} : b_1, ..., b_{V-1}) + d(a_T : b_V)
\right\}$$
Experiments

- Better results obtained for all mono corpora over those distances
- Success rates (best robust result for each method given all possible propagations, pruning levels and pitch representations)

<table>
<thead>
<tr>
<th></th>
<th>VAR</th>
<th>MIREX</th>
<th>Pascal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shasha</td>
<td>75</td>
<td>43</td>
<td>75</td>
</tr>
<tr>
<td>Selkow</td>
<td>69</td>
<td>43</td>
<td>95</td>
</tr>
<tr>
<td>Valiente</td>
<td>69</td>
<td>43</td>
<td>75</td>
</tr>
<tr>
<td>Partial</td>
<td>75</td>
<td>46</td>
<td>95</td>
</tr>
</tbody>
</table>

Note: not all best results are obtained for the same prop. and pruning level for each method.
Use available inf. to compare

- Integrate into the same tree harmony, rhythm and pitch
- Compare labels containing also tonal analysis

Subst cost= linear combination of present properties (pitch, T.F., degree)

Promising early results
Use available inf. to compare

- Lack of tagged corpora
- Proof of concept
  - using melisma to analyze and integrate it
  - need a command that outputs chords + tonality + time
  - it works with quantization in “pips” = 35 ms units
  - alter source code so that all this information is printed: chord + tonality + time in ticks

#Kostka-Payne Roman Numeral Analysis - DRizo version:
| C#m:@0 VI64@0 VI64@1155 | VI6@1750 | VI64@3535 VI64@4760 | VI6@5320 | VII@7140 ............
Comparison with others

- Obtained implementations of other similarity methods:
  - Suyoto & Uitdenboderg: fanimae (n-grams based) (mono)
  - A.Pinto: graph based (mono)
  - Grachten: Narmour (waiting...) (mono)
  - J.Bernabeu implements PROMS (poly)
PROMS

- Implementation integrated with Similarity Computation Framework

- Current output: index=n shows the number of the piece (last number = coincidence rate)

<table>
<thead>
<tr>
<th>Index</th>
<th>Hits</th>
<th>Misses</th>
<th>Bar</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>17</td>
<td>12</td>
<td>#0</td>
<td>19/19</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>16</td>
<td>#4</td>
<td>3/19</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>16</td>
<td>#0</td>
<td>3/19</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>18</td>
<td>#1</td>
<td>1/19</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>18</td>
<td>#9</td>
<td>1/19</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>18</td>
<td>#0</td>
<td>1/19</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>18</td>
<td>#5</td>
<td>1/19</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>18</td>
<td>#0</td>
<td>...</td>
</tr>
</tbody>
</table>

Current similarity values: sum of rates, maximum rate. Lemström suggests to “boost” low density pieces when compared with high density ones. Any better idea?
Polyphonic methods

- Only 3 real polyphonic methods
  - real = that compare directly polyphonic music
  - Lemström P1-3, PROMS, Our multiset trees
  - Now we have combined P-n and trees. Aim: combine also PROMS

- Other “non-real” polyphonic methods
  - compare mono to poly: e.g. Ferraro quotiented trees (actually are sequences of sets of notes)
  - 1st step: poly to mono reduction (e.g. skyline)
  - good results using this method with our corpora and our methods
Launched

1) Integration with Pertusa’s transcription system

- MIDI to audio with timidity
- audio to midi with transcriptor
- similarity using all existing methods (strings, graphs,...)
  - trees: need time: use original MIDI info / use melisma (:/) to get metrical structure
- compare results

- Ready: corpus of audio covers
2) J. Bernabeu k-testables to compare music

- we use to compare trees to strings
- let’s compare k-testables to n-grams
- C. Pérez implemented this comparison
- Pending: use global methods of P. J. Ponce de León to compare music also
Tonal analysis

- After summer’08 stay at IRCAM
- OpenMusic Implementation reviewed with P.R. Illescas
- Designed an intermediate file that exports the analysis graph from OpenMusic to be used by J.Oncina for Consolider (next slide)

- pending: incorporate manual analysis from MusicXML files
**Chord 2:** (7 10 2)

**possibilities:** (cmin v min d) (cmin v min t) (dmin iv min s) (gmin i min t) (ebmaj iii min s) (ebmaj iii min t) (ebmaj iii min d) (fmaj ii min s) (bbmaj vi min s) (bbmaj vi min t)

**expected:** (gmin i min t)

**our analysis:** (gmin i min t)

---

**Chord 3:** (7 10 2)

**possibilities:** (cmin v min d) (cmin v min t) (dmin iv min s) (gmin i min t) (ebmaj iii min s) (ebmaj iii min t) (ebmaj iii min d) (fmaj ii min s) (bbmaj vi min s) (bbmaj vi min t)

**expected:** (gmin i min t)

**our analysis:** (gmin i min t)

........