TREES AND COMBINED METHODS FOR MONOPHONIC MUSIC SIMILARITY EVALUATION

David Rizo, José M. Iñesta
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante, Spain
{drizo,inesta}@dlsi.ua.es

ABSTRACT

This abstract describes the four methods presented by us with the objective of obtaining a good trade-off between accuracy and processing time [3]. Three of them are based on a summarization of the input musical data: the tree representation approach [5, 6] (*UA_T-RI2*, and *UA_T3-RI3*), and the quantized point-pattern representation [1] (*UA_PR - RI4*). The fourth method is an ensemble of methods [4] (*UA_C-RI1*). The summarization methods are expected to be faster than approaches dealing with raw representations of data. The ensemble combines different approaches trying to be more robust and are expected to give equal or better accuracy than the summarization methods. Thousands of different parametrizations of those methods are possible. The parameters of the presented methods are chosen based on previous experiments.

1. TREE REPRESENTATION OF MELODIES (UA_T AND UA_T3)

Music pieces can be represented by symbolic structures such as strings or trees containing the sequence of notes in the melody. A melody has two main dimensions: rhythm (duration) and pitch. In linear representations, both pitches and durations are coded by explicit symbols, but trees are able to implicitly represent time in their structure (the shorter a note the deeper it is in the tree), making use of the fact that note durations are multiples of basic time units in a binary (sometimes ternary) subdivision (see Fig. 1). This way, trees are less sensitive to the codes used to represent melodies, since only pitch codes are needed to be established and thus there are less degrees of freedom for coding.

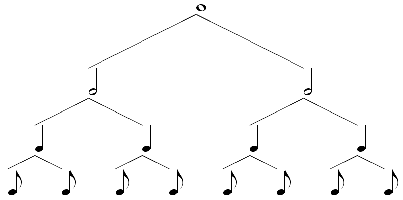
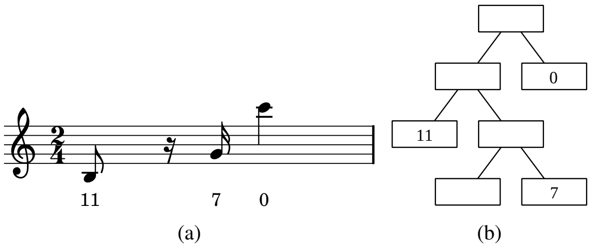


Figure 1: Duration hierarchy for note figures. From top to bottom: whole (4 beats), half (2 beats), quarter (1 beat), and eighth (1/2 beat) notes.

For representing the note pitches in a monophonic melody s as a string, symbols σ from a pitch representation alphabet Σ_p are used: $s \in \Sigma_p^*, s = \sigma_1\sigma_2...\sigma_{|s|}$. Several encodings can be found in the literature trying to solve two issues: the transposition and interval invariance [7], and the level of encoding precision tied to the error tolerance. For the presented algorithms based on tree representations, the *high definition contour* pitch representation p_{hdc} has been chosen for being interval invariant and in the middle of the trade-off between error tolerance and representation precision.

Definition 1.1 *Let p_{abs} be the MIDI pitch, and n_i the i-th note:*

$$\Sigma_{p_{hdc}} = \begin{cases} `+2' & \textit{if}\, p_{abs}(n_i) > p_{abs}(n_{i-1}) + 4 \\ `+1' & \textit{if}\, p_{abs}(n_i) \leq p_{abs}(n_{i-1}) + 4 \\ & \wedge p_{abs}(n_i) > p_{abs}(n_{i-1}) \\ `-2' & \textit{if}\, p_{abs}(n_i) - 4 < p_{abs}(n_{i-1}) \\ `-1' & \textit{if}\, p_{abs}(n_i) - 4 \geq p_{abs}(n_{i-1}) \\ & \wedge p_{abs}(n_i) < p_{abs}(n_{i-1}) \\ `0' & \textit{otherwise}. \end{cases}$$

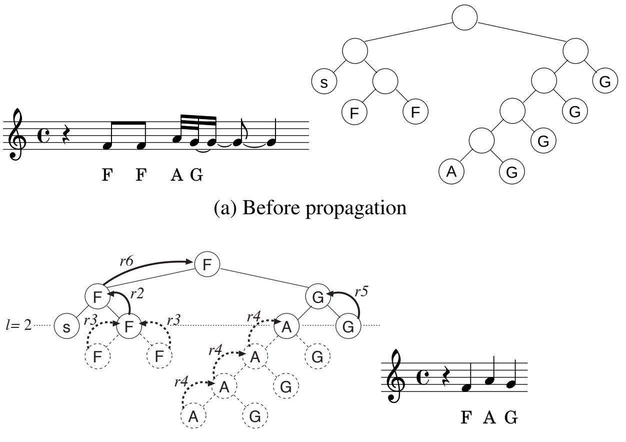


(a)

(b)

Figure 2: Melody and a tree representation before propagation using as pitch representation the *pitch class*

In the proposed approach, each melody bar is represented by a tree, $t \in T_{\Sigma_p}$ (the set of trees that can be made with the labels in Σ_p). The level of a node in the tree determines the duration it represents (see an example in Fig. 2). The root (level 1) represents the duration of the whole bar, the two nodes in level 2 the duration of the two halves of a bar (in this case, two quarter notes), etc. In general, for a binary meter, nodes at level i represent duration of a $1/2^{i-1}$ of a bar ($1/3^{i-1}$ for a ternary meter). Therefore, during the tree construction, nodes are created top-down when needed and guided by the meter, to reach the appropriate leaf level to represent a note duration (notes are split to accommodate node durations). At that moment, the corresponding leaf node is labeled with the pitch representation symbol, $\sigma \in \Sigma_p$. Once the tree has been built, a bottom-up propagation of the pitch labels is performed to label all the internal nodes. Several propagation schemes have been proposed (based on melodic analysis: -harmonic tones, passing tones, ...-, empiric, always left, always right) [3] (see Fig.3).



(a) Before propagation

(b) After propagation

Figure 3: Bottom-up propagation of labels using the *empiric* propagation scheme.

For the presented algorithms, we have chosen the *melodic* propagation. The rules for this propagation are based on a melodic analysis [2]. All the notes are tagged either as *harmonic tones* for those belonging to the current harmony at each time, or as *non-harmonic tones* for those ornamental notes. Harmonic notes have always priority for propagation and when two harmonic notes share a common father node, propagation is decided according to the metrical strength of the note (the stronger the more priority), depending on its position in the bar and the particular meter of the melody. Note that each bar may have a different time signature. Notes have always higher priority than rests. Eventually, all the internal nodes are labeled, yielding the tree t_i that codes the i-th bar of the melody. This process is repeated for all the bars in the melody.

At this point, all the bar trees: $t_1, t_2, ..., t_{|M|}$, where $|M|$ is the length of the melody in bars, are linked to a common root, building up a forest, $\sigma(t_1 t_2 ... t_{|M|}) \in T_{\Sigma_p}$, where that common root is labeled with the root of the first tree, corresponding to the first harmonic tone of the melody, after the melodic analysis performed for the label bottom-up propagation.

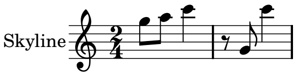
After having all the nodes tagged the trees are ready to be compared in order to give a similarity value between the represented musical works. Several similarity measures between trees can be found in the literature [3]. For this contest we have selected the Selkow edit distance [8].

The tree edit distances have a high temporal computation cost that depends mainly on the number of levels of the tree and the number of leaves. By reducing the tree depth the performing times are dramatically reduced. Under the assumption that the most important notes in the melody are propagated bottom-up, the upper tree levels contain the more important notes in the melody. We use this assumption to prune the tree removing all nodes lying below a given pruning level l (see $l = 2$ in Fig. 3). The smaller the pruning level the faster the methods with the cost of lower precision. For the contest we have selected the pruning level $l = 2$ (*UA_T-RI2*), and $l = 3$ (*UA_T3-RI3*).

2. BAR-SPECIFIC, QUANTIZED POINT-PATTERN REPRESENTATION (UA_PR)

Originally designed for locating occurrences of excerpts of polyphonic songs in music databases [1], we have adapted it to give a numerical score of the similarity between two musical works, polyphonic or monophonic.

In order to represent a melody, each note is encoded with a pair $< qtime, pitch >$ where $qtime = \frac{onsettime}{q}$ and q is the quantization resolution (all values are relative to the song resolution in ticks). This quantization leads to groups of pairs with the same $qtime$. For the contest, we have chosen the value of a quarter for q. Sample melody using two quantizations:



$q = quarter$: {[0,69], [0,67], [1,72], [2,55], [3,72]}
$q = 8th$: {[0,67], [1,69], [2,72], [5,55], [6,72]}

Figure 4: Bar-specific, quantized point-pattern representation

For performing the comparison, both matrices are constructed for the target song and the query of dimensions $bar \times qtime \times pitch$. The alignment (in these three dimensions) of the two matrices with the highest number of coincidences is computed, and the similarity value is the number of coincidences normalized by the maximum number of bars.

3. STRING METHODS

The string methods have been used in the ensemble approach presented. On the contrary of tree representations, strings require an explicit representation of rhythm. Several considerations have to be taken into account to select the most adequate rhythm encoding. The invariance against changes in meter dealt with relative representation as opposed to the absolute codes, and the level of encoding precision that has impact on the error tolerance. For the presented approach we have used the absolute onset time encoded in units relative to the quarter note, i.e., a quarter note is represented as a '1', a half note as a '2', and an eighth as '0.5'.

The string representations has many more aspects that must be taken into account, they are:

- Whether include or not harmonic tones and rests. As shown above in the trees approach, the non-harmonic tones are discarded in the propagation scheme, so a possibility in string representations is to remove these notes. Rests, if are present in the input score, can also been omitted.

- Having two dimensions to be represented, these two dimensions can be represented *coupled* or *decoupled*. For the *coupled* representation a single symbol for each note is used containing both pitch and rhythm. This implies that for the edit distance used, the substitution cost must combine the similarity of those two dimensions of pitch and rhythm. For *decoupled* representations, a symbol is either a pitch or a rhythm code, represented by different alphabets, in such a way that it makes not sense to compare pitch and rhythm symbols. The advantage of *decoupled* representations is that it makes possible to recognize similar rhythm subsequences in a part of the song and similar pitch subsequences in other part.

- Other thing to consider is the edit distance to be used: either local or global, and the editing costs used for those distances.

- Finally, for the *coupled* representations, the combination constant k of pitch and rhythm substitution cost: the replace cost of pitch p and rhythm r is a linear combination of the substitution cost of both.

We have not sent any string method to the contest, but they have been included in the ensemble that is described below. Two versions of the strings have been used, one *coupled* and other *decoupled*. In both cases the rhythm absolute time r_{tabs} and $k = 0.9$ have been used. For coupled strings the pitch p_{hdc} is used, for decoupled strings the interval pitch p_{itv} has been chosen.

4. ENSEMBLES

This edition of the contest has had no training phase in which base the most adequate setup of the combination of methods. Thus, we have use knowledge from previous experiments on other corpora to construct the ensembles.

Two decisions must be taken to construct an ensemble: which methods have to be included and how to combine the individual results of those methods.

In order to select the most adequate combination of methods and parametrization we have used the approach presented in [4], i.e., select the methods giving the most diverse results with the best performing rates. The included methods have been the above described trees with $l = 2$, the bar-specific, quantized point-pattern representation, and the two string representations.

5. EXPERIMENTS AND RESULTS

The system sent to the contest work the same way for the four methods presented. For the first invocation, the full corpus is read, encoded in any of the used representation, and then saved as a text file. Note that this is not a real indexing process, but a trick to avoid the encoding of the whole corpus for each invocation. Then, for each query, this text file is read, the query is encoded in the given representation, and compared to each song in the corpus, ranking the results by the normalized similarity value.

The results confirm the expected: the fastest methods among almost all presented to the contest have been those based on summarization (UA_T, UA_T3, UA_PR), and the ensemble approach (UA_C) has reported the best accuracy among those presented by us. The songs used seem to be too short for the trees with $l = 2$ to compare similarity.

6. BIBLIOGRAPHY

7. REFERENCES

[1] Michael Clausen, Ronald Engelbrecht, D. Meyer, and J. Schmitz. Proms: A web-based tool for searching in polyphonic music. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2000.

[2] Plácido R. Illescas, David Rizo, and José Manuel Iñesta. Harmonic, melodic, and functional automatic analysis. In *Proceedings of the 2007 International Computer Music Conferrence*, volume I, pages 165–168, 2007.

[3] David Rizo. *Symbolic music comparison with tree data structures.* PhD thesis, Universidad de Alicante, (to be defended) 2010. `http://www.dlsi.ua.es/gent/drizo/ThesisDRizoDraft.pdf`.

[4] David Rizo, Kjell Lemström, and José Manuel Iñesta. Ensemble of state-of-the-art methods for polyphonic music comparison. In *Proceedings of the WEMIS Workshop, ECDL 2009*, pages 46–51, Corfu, Greece, October 2009.

[5] David Rizo, F. Moreno-Seco, and José Manuel Iñesta. Tree-structured representation of musical information. *Lecture Notes in Computer Science - Lecture Notes in Artificial Intelligence*, 2652:838–846, 2003.

[6] David Rizo, F. Moreno-Seco, José Manuel Iñesta, and L. Micó. *Efficient search with tree-edit distance for melody recognition*, chapter 14, pages 218–244. Centre de Visió per Computador, 2006.

[7] Eleanor Selfridge-Field. Conceptual and representational issues in melodic comparison. In Walter B. Hewlett and Eleanor Selfridge-Field, editors, *Melodic Similarity: Concepts, Procedures, and Applications. Computing in Musicology*, volume 11, chapter 1, pages 223–230. MIT Press, 1998.

[8] Stanley M. Selkow. The tree-to-tree editing problem. *Information Processing Letters*, 6(6):184–186, 1977.