CLUSTERING SYMBOLIC MUSIC USING PARADIGMATIC AND SURFACE LEVEL ANALYSES

Anna Pienimäki and Kjell Lemström
University of Helsinki
Department of Computer Science
{Anna.Pienimaki, Kjell.Lemstrom}@cs.Helsinki.FI

ABSTRACT

In this paper, we describe a novel automatic cluster analysis method for symbolic music. The method contains both a surface level and a paradigmatic level analysing block and works in two phases. In the first phase, each music document of a collection is analysed separately: They are first divided into phrases that are consequently fed on a harmonic analyser. The paradigmatic structure of a given music document is achieved comparing both the melodic and the harmonic similarities among its phrases. In the second phase, the collection of music documents is clustered on the ground of their paradigmatic structures and surface levels. Our experimental results show that the novel method finds some interesting, underlying similarities that cannot be found using only surface level analysis.

1. INTRODUCTION

The area of Music Information Retrieval (MIR) has extended to contain several tracks. One of these is music classification. Here we consider an unsupervised version of classification, cluster analysis. We will call the pieces of music *documents* and the set of all the given documents *collection*.

The aim of the cluster analysis is to cluster the most similar objects close to each other. The similarity between clustered objects is most commonly measured using distance functions, which define objects to be the more similar to each other the smaller the distance is between them. In MIR clusters are used, for example, in playlist generation or style identification tasks. Musicologists, on the other hand, may use the results of cluster analysis when preprocessing their analysis material, such as material collected during a field study.

Music can be seen as a structural phenomenon consisting of many levels [5]. The surface level, distinct notes or small combinations of them, is quite easily approached and therefore widely studied in computer-assisted musicology and MIR. Indeed, in many cases there is no need for deeper, musicological analysis. For instance, amateur listeners recognise similarities between pieces of music mainly by surface level characteristics such as melodic and rhythmic patterns. For musicologists, the surface level analysis is necessary but by no means sufficient. However, there are only a few studies (see e.g. [1]) that algorithmically analyse deeper structural levels, such as the paradigmatic level, i.e., a rough segmentation of the music document into phrases [6].

This paper introduces a new cluster analysis method for symbolic music data based on both paradigmatic and surface level analysis. Each document of the collection is first analysed and encoded using its paradigmatic structure, melody, and harmony as its parameters. Then the whole collection is clustered using a distance measure that calculates both the paradigmatic and the surface level similarity among the music documents.

Background. In data clustering the main three problems are how to describe the data items comparably, what kind of distance measure is suitable for describing the similarities between data items, and how to find the most representative item for a cluster. Often data items are represented by n-dimensional vectors that are compared by using some nicely behaving distance measure (such as some *metric*).

When dealing with (symbolic) music, selecting a suitable feature to be extracted may not be obvious. The data items may also be complex which makes it difficult to find a suitable distance measure. Moreover, when dealing with non-numerical values, each representative of a cluster has to be an existing data item, not any artifact such as the mean value of the cluster.

In the literature, one can find two approaches for clustering music. The first, which we call *paradigmatic clustering*, gets individual documents as input (in the form of pre-defined phrases or analysed document excerpts of approximately equal length) and aims at finding some inner structure (see e.g. [1]). The phrases are described by using several features such as the melodic curve, the intervals of the melody, and the rhythmic relationships. The other approach, *collection clustering*, aims at making clusters of documents in a given collection (see e.g. [2]). These methods often work on secondary data, such as statistical

characteristics of the melody.

Our novel method, to be described in the following sections, combines ideas of both of these approaches. It first segments each document of the collection into phrases; the phrase similarity is based both on the melodic line and on the harmonic analysis of the excerpt containing the phrase at hand. The paradigmatic structure of the documents is obtained by cluster analysis using information collected during the former phase. To describe the documents we use adjacency lists; each such list is associated with a document and stores results of paradigmatic and surface level analyses of the corresponding document. The whole collection can then be clustered by using the adjacency lists.

2. CLUSTERING PARADIGMATIC LEVEL USING SURFACE LEVEL INFORMATION

Let us now describe how we analyse individual documents. This includes describing the surface level and then forming paradigmatic clusters based on the surface level descriptions. The adjacency list storing these results is described in Section 3.

Temperley has introduced heuristics for segmenting monophonic music into phrases [7]. In order to apply them here, we need a melody extraction method because our collection contains mainly polyphonic documents. For the moment, we use a simplistic approach that considers notes with the highest pitch values as melody.

Having applied Temperley's phrase segmentation, we have the *melodic phrases*. For each melodic phrase, we also need the associated polyphonic context, the *harmonic phrase*. Having extracted these out of the original document, we analyse them using Temperley's harmonic analysis heuristics [7]. Because the length of the harmonic phrases may vary, we compress them by replacing runs of any symbol with a single symbol. For instance, string $I\ IV\ V^6\ V^6\ I$ is replaced by $I\ IV\ V^6\ I$.

Let $x = x[1], \ldots x[n]$ be a string of length n on an alphabet Σ. We use such strings to represent extracted phrases: Let p be such an extracted phrase. Then $p^M = p^M[1], \ldots, p^M[n]$ and $p^H = p^H[1], \ldots, p^H[n]$ represent its melodic and harmonic parts, respectively. If there are k extracted phrases for some document d, we form a similarity matrix S^d of size $k \times k$, such that the cell $S^d_{a,b}$, for $1 \leq a, b \leq k$, is given by the formula:

$$S^d_{a,b} = c * D^H(p^H_a, p^H_b) + (1 - c) * D^M(p^M_a, p^M_b), \quad (1)$$

where D^M and D^H are melodic and harmonic edit distances, respectively, as defined below. Factor c ($0 \leq c \leq 1$) adjusts the weights of the components.

The harmonic distance, D^H, on two strings p and p' of lengths $|p| = n$ and $|p'| = m$ is calculated by the following recurrence resembling the conventional edit distance:

$$D^H_{00} = 0 \quad (2)$$

$$D^H_{ij} = \min \begin{cases} D^H_{i-1,j} + 1; \\ D^H_{i,j-1} + 1; \\ D^H_{i-1,j-1} + (\text{if } p[i] = p'[j] \text{ then } 0 \\ \qquad \text{else } \mathcal{D}(i,j)). \end{cases}$$

The distance between strings p and p' can then be found reading the value of the entry D^H_{nm}. The additional part, $\mathcal{D}(i,j)$, takes into account musicological findings. It is defined as follows. $\mathcal{D}(i,j) =$

$$\min \begin{cases} a_1, \text{ if } p[i], p'[j] \text{ are inversions of same chord} \\ a_2, \text{ if } p[i], p'[j] \text{ are different chords of same function} \\ a_3, \text{ if } p[i], p'[j] \text{ are major and minor forms of same chord} \\ 1, \text{ otherwise.} \end{cases}$$

The coefficients a_1, a_2, and a_3 ($a_1 \leq a_2 \leq a_3$) enable weighting between functional differences of chords. The functions of the chords are defined traditionally: the *tonic* function includes chords I and VI, the *subdominant* IV and II, the *dominant* V and VII. The chord III belongs in class *others*.

The melodic edit distance, D^M, in Equation 1 is calculated by using the straightforward transposition invariant edit distance, D_N, as defined in [4].

Paradigmatic Level Clustering. In paradigmatic level, we cluster together phrases that can be seen as variants of one common phrase. The result is often represented as a string of paradigm symbols, such as *ABCCA*. Thus, paradigmatic level suggests also a rough segmentation for the document.

To find the cluster, we iterate the similarity matrix $S^d_{a,b}$ (Equation 1). Between each iteration, the similarity matrix is updated using the complete link approach (see e.g. [3]) that tends to build spherical clusters. The iteration is halted when the values in the similarity matrix exceed a predefined threshold value α. The paradigms are labeled in the alphabetic order starting with A.

3. CLUSTERING MUSIC COLLECTION

The results of the analyses described above are stored into an adjacency list (Fig. 1). The main body of this structure, which is merely an array of header cells, is conceptually divided into two parts: paradigmatic and surface level parts. The header cell of the paradigmatic part contains a link to the formed paradigm string. The surface level part contains varying amounts of header cells, one for each paradigm. For each such header cell, there is a link to a list containing two strings that represent the associated harmonic and melodic phrases. When a paradigm has several variations, the phrase with the lowest distance to all the other variations is opted for the representative phrase of the paradigm.

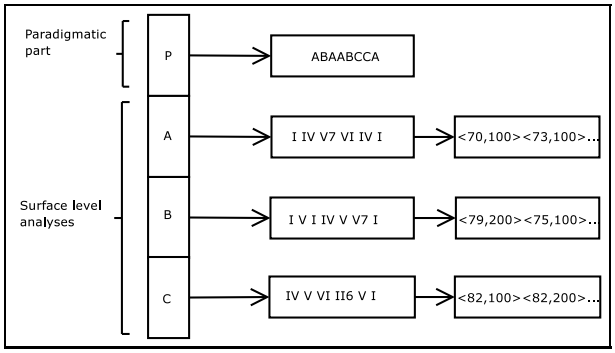

Figure 1. Adjacency list corresponding to document.

Recall that when we measure the similarity of two music documents, we want to combine the similarity of their paradigmatic structures with the similarities of the corresponding surface levels. Once we are able to do that, we can construct a similarity matrix (analogous to that of Equation 1) for all the documents in a given collection. Then we can cluster the whole collection using the constructive hierarchical algorithm that was already used for paradigmatic clustering. The resulting dendrogram will give us the hierarchical structure of the collection.

To that end, let d and d' be two analysed documents and let $p = ABCB$, $p' = AABA$, and s, s' denote the associated paradigm and surface strings, respectively. The straightforward way to couple d and d' together would be to align paradigms sharing the same label and calculate the corresponding surface level (phrase) similarities. Naturally, this may not be the optimal choice because the labeling process is consistent only over a single document. When having a closer look at the paradigm strings of p and p' above, one can hypothesise that coupling where B and C in the first string are aligned with A and B in the second, respectively, may result in a more appropriate interpretation. Obviously the optimal coupling depends on the corresponding surface level structures s and s', not on the paradigm labels.

Since the optimal coupling is computationally very expensive, we have implemented a greedy version that results in nearly optimal coupling in most of the cases. For assessing the similarity of two paradigm strings, we have implemented three distance measures based on the edit distance framework. The first, denoted by D^1, is very straightforward, yet in many cases effective. It applies conventional edit distance (with deletion, insertion, and substitution operations) to the paradigm strings without any relabeling. The second, D^2, is based on Hamming distance (the only allowed editing operation is substitution) and behaves more nicely in the cases where the two strings clearly share a similar structure, but one string is much longer than the other. The third, D^3, works as D^2 but also relabels the paradigm string p' accordingly with p by using the alignment given by the greedy coupling.

Let p and s denote the paradigm and surface level strings of some document d, and n the number of documents in collection C. We form a similarity matrix S^C of size $n \times n$, such that the cell $S^C_{i,j}, 1 \leq i, j \leq n$, is given by the formula:

$$S^C_{i,j} = c_2 * D^S(s_i, s_j) + (1 - c_2) * \min \begin{cases} D^1(p_i, p_j) \\ D^2(p_i, p_j) \\ D^3(p_i, p_j), \end{cases} \quad (3)$$

where c_2 ($0 \leq c_2 \leq 1$) is a parameter used to adjust the relative weight between paradigmatic and surface level structures. Let $|s|$ be the number of paradigms in document s, and $m = \max(|s|, |s'|)$. Now, $D^S(s, s')$ gives the surface level similarity and is defined by the formula:

$$D^S(s, s') = \sqrt{\sum_{i=1}^{m} (S^d_{s_i, s'_i})^2}, \quad (4)$$

where S^d is the surface distance defined in Equation 1. The paradigms compared, s_i and s'_i, are selected using the greedy coupling. If $|s| \neq |s'|$ the paradigms without pairs are compared to null strings.

The minimum operation in Equation 3 opts for one of the possibilities for paradigmatic structure similarities. While the intuition of D^1 should be obvious (see e.g. [4]), the idea behind D^2 (and D^3) requires further explaining. This is done in the following paragraph.

Ignoring Repetitions. Let us have two strings p and p' of lengths $|p| = n$ and $|p'| = m$, such that $n - m = \ell > 0$. When applying the conventional edit distance, D^1, to them, $D^1(p, p') \geq \ell$ although the structure of p may just be a prolonged version of p', like in case: $p = ABCABCABC$ and $p' = ABC$.

Now, if we consider the shorter string as a search pattern to be searched for in the longer string, we can easily see that p' occurs exactly in p at positions 1, 4, and 7. This suggests that p' and p are paradigmatically much closer than $D^1(p, p') = 6$ would hint. It is well-known that the edit distance framework can be straightforwardly adapted to the pattern matching case. Now, instead of finding the minimum value at the bottom row of the dynamic programming table computing the matching process, we collect $\lceil |p|/|p'| \rceil$ minimum values. The mean value of this set is given as the distance between the two strings.

In order to obtain meaningful results in doing so, we need to apply the Hamming distance. This is due to the well-known fact that the adjacent values in the dynamic programming table vary at most by one when allowing insertions, deletions and substitutions. If done so, in the resulting difference one perfect occurrence may be accounted several times, giving a small score although the structures would have been rather dissimilar.

As for an example of our measure D^2, let us consider p and p' as given above. The bottom row of the dynamic programming table would be 3, 3, 0, 3, 3, 0, 3, 3, 0 and $\lceil \frac{|p|}{|p'|} \rceil = 3$. Because there are three zeros in the bottom row (three exact occurrences of the pattern), $D^2(p, p') = 0$. In this case, the result describes intuitively better the similarity of p and p' than $D^1(p, p') = 6$. At this point,

the functioning of D^3, as explained above, should have become obvious, as well.

4. EXPERIMENTAL RESULTS

Let us now describe briefly our implementation, the influence of coefficients used, the test sets, and the main results obtained. As mentioned in Section 2, we used segmentation and harmonic analysis modules based on Temperley's heuristics [7]. Implementations of them can be found freely available for non-commercial use [1]. We implemented the melody extraction, clustering, and distance calculation algorithms as Perl scripts. The data flow between various scripts was managed by a shell script.

We studied the behaviour of the method using two test sets containing documents in MIDI format. The first set contained 48 short fragments of classical pieces. This set was used mainly for evaluating the functionality of the method. For more thorough analysis of the method we built the second test set by choosing 145 classical pieces randomly from the Mutopia [2] database. The size of the second set was intentionally kept as small as possible, because of the huge amount of handiwork needed when analysing the results of the clustering.

First we studied the paradigmatic clustering phase. We experimented on various threshold values α. At each considered value, we generated a list of adjacency lists and analysed the quality of paradigmatic clustering. We found that the threshold value 20 gave the best paradigmatic clustering result in this data set.

We experimented also on the coefficient c of Equation 1. In the first experiment the value of c in paradigmatic phase was set in $\frac{2}{3}$. We found that the harmonic analysis had difficulties with surface level chromaticism and the results skewed the paradigmatic clustering, as well. In the second experiment we set the value of c in $\frac{1}{4}$. In this case, the problems of the chromaticism did not dominate the paradigmatic clustering as strongly as in the first experiment.

The coefficients a_1, a_2, and a_3 were set in 0.25, 0.5 and 0.75, respectively. In future, we will evaluate a wider range of values for them.

In the collection clustering phase we experimented on coefficient value c_2. We used c_2 in testing the importance of paradigmatic structure setting c_2 to be 1 and thus eliminating its influence. We found that when setting the value of c_2 very high, the resulting hierarchical structure was constructed mostly on the basis of occasional similarities on the surface level. These similarities are of course interesting in such research problems that aim at finding, for instance, similar melodic patterns from music documents. In our case, however, these similarities, especially when occurring without wider musicological context, are not very interesting.

When the value of c_2 was set low thus stressing the influence of the paradigmatic level similarities the resulting structure was altered. We found that certain documents that were located near to each other when the surface level similarity was stressed, were in the same subcluster also when stressing the paradigmatic level similarity. In this case we found both paradigmatic and surface level similarities between the documents when examining the adjacency lists of the documents. In many cases, however, the similarity between certain documents was noticeable in the hierarchical clustering structure only when stressing the paradigmatic level similarity. This behaviour was characteristic of documents containing especially Bach's compositions. This kind of similarity could not be found by examining only the surface level of the documents.

[1] http://www.link.cs.cmu.edu/music-analysis/

[2] http://www.mutopiaproject.org/

5. CONCLUSION

We described a novel automatic analysis method based on paradigmatic and surface level similarity of music represented in symbolic form. Our experimental results on the method were encouraging, given the simplicity of it. Because of the modular structure, our method is easily customised: other surface level descriptions, such as statistical parameters, can be accommodated by adding a term into Equation 1.

6. REFERENCES

[1] Cambouropoulos E. and Widmer, G., "Automatic motivic analysis via melodic clustering", *Journal of New Music Research 29*(4), pp. 303–317, 2000.

[2] Eerola, T., Järvinen, T., Louhivuori, J., and Toiviainen, P. "Statistical features and perceived similarity of folk melodies", *Music Perception* 18, pp. 275–296, 2001.

[3] Hand, D., Mannila, H., and Smyth, P., *Principles of Data Mining*. MIT Press, Cambridge, Mass., 2001.

[4] Lemström, K. and Ukkonen, E., "Including interval encoding into edit distance based music comparison and retrieval", *Proc. AISB Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science*, Birmingham, UK, pp. 53–60, 2000.

[5] Lerdahl, F. and Jackendoff, R., *A Generative Theory of Tonal Music*. MIT Press, Cambridge, Mass., 1983.

[6] Nattiez, J.-J., *Music and Discourse – Toward a Semiology of Music*. Princeton University Press, 1990.

[7] Temperley, D., *The Cognition of Basic Musical Structures*. MIT Press, Cambridge, Mass. 2001.