A Fuzzy-Clustering Based Approach for Measuring Similarity Between Melodies

Brian Martínez[1], and Vicente Liern[2]

[1]Conservatorio Superior de Música, Valencia, Spain.
[2]Dep. Matemáticas para la Economía y la Empresa,
Universidad de Valencia, Spain.
brian.martinez@csmvalencia.es,vicente.liern@uv.es

Abstract. Symbolic melodic similarity aims to evaluate the degree of likeness of two or more sequences of notes. In this work, we propose the use of fuzzy c-means clustering as a tool for the measurement of the similarity between two melodies with a different number of notes. Moreover, we present an algorithm, FOCM, implemented in a computer program written in C♯ able to read two melodies from files with MusicXML format and to perform the clustering to calculate the dissimilarity between any two melodies. In addition, for each iteration step in the convergence process of the algorithm, a family of intermediate states (transition melodies) are obtained that can be used as new thematic material. This last feature, could be especially useful in the near future, as a complement in computer-aided composition.

Keywords: Fuzzy Clustering, Simbolic Melodic Similarity, Computer-Aided Composition.

1 Introduction

Symbolic melodic similarity is fundamental in the field of computer-aided composition [1, 13]. The measure of the similarity/dissimilarity between melodies is a key factor both in defining transitions between two different melodies and to generate new melodic material from an already preexisting melody [11]. In this paper, we propose a procedure to measure the similarity between two melodies by using an algorithm based on fuzzy clustering.

Our starting point will be the characterization of musical notes as points in a metric space, where the coordinates represents musical characteristics. In this way, a melody would be an ordered sequence of notes. Measuring the dissimilarity between two monophonic melodies of equal number of notes can be made by comparing, one by one, each note of the first melody with a note of the same order in the second. However, for our purpose we would need to be able to establish a generic comparison mechanism allowing the measurement of the dissimilarity of two (monophonic or polyphonic) melodies of different number of notes. For this, we will use an algorithm based on fuzzy c-means clustering.

The purpose of that clustering would be to establish to what extent the notes of a first melody are related to the notes of a second one. The purpose of that clustering would be to establish to what extent the notes of two different melodies are related. After the clustering, we will be able to calculate a global difference between the melodies (dissimilarity) aggregating the partial distances weighted by their corresponding membership coefficient. In the fuzzy logic context, the membership functions are the extension of the characteristic set functions [16]. While characteristic functions take values 0 or 1, membership functions can take any value between 0 and 1. Therefore, the membership coefficients express the membership degree of an element to a cluster [16, 15].

Subsequently, in the comparison of the general dissimilarity, we will take into account the order of the notes in each melodic sequence. For this, we will use neighborhood functions. These functions will allow us to define a comparison in which the clustering of the notes is influenced by their position within the sequence defining the melody.

In order to verify the utility of our proposal, we present an algorithm, FOCM, implemented in a computer program written in C♯. This algorithm will allow us to read two melodies from files in MusicXML format and to perform the clustering to calculate the dissimilarity between them. In addition, for each iteration step in the convergence process of the algorithm, a family of intermediate melodies will be obtained that can be used as new thematic material. This last feature could be especially useful in the near future, as an aid in computer composition. For this reason, in the last section we provide an example, in which the number of intermediate melodies created by our method is shown. As an instance, we present one of the intermediate melodies obtained from the measure of dissimilarity between two passages.

2 Preliminary Concepts

A musical note determined by k characteristics (pitch, intensity, duration, timbre, etc.) can be expressed as a vector in $\mathbb{R}^q$, where $q \leq d$. Of course, each characteristic does not have to correspond to a single coordinate. For example, in [7, 8] pitch is defined as a fuzzy set [16],

$$\tilde{P} = \big\{(f, \mu_{\tilde{P}}(f)), \quad f \in [F_0, F_1]\big\}, \tag{1}$$

where f represents the frequency in Hz. and $\mu_{\tilde{P}}(f) \in [0, 1]$ is the membership degree of f to a note in a given tunning system. In this case, the fuzzy pitch would be given by two coordinates $(f, \mu_{\tilde{P}}(f))$.

The most simple way to represent a musical note is by setting $q = 2$, the pitch and the duration, and establishing two bijections from the pitch and the duration of the note $\mathbf{x} \in \mathbb{R}^2$. However, it is possible to work with a higher number of dimensions in order to represent more accurately the characteristics of music. New properties belonging to the requirements of other kinds of music or styles [14], like *Non Western-tradition music*, *Computer-Generated Music* or *Electroacoustics* could be easily assimilated as extra dimensions.

As usual, the distance in cents between two notes whose frequencies are f_1 and f_2 can be easily calculated in cents [3, 8] by means of the expression

$$d(f_1, f_2) = 1200 \times \left|\log_2\left(\frac{f_1}{f_2}\right)\right| \quad \text{cents.} \tag{2}$$

According to [12], the MIDI protocol defines a midi-pitch of a note by a integer number comprised on a range $[0, 127]$, being central $C_4 = 60$ and reference $A_4 = 69$. For the equal temperament of 12 notes [3], there are 100 cents of difference between two notes separated by one midi-pitch number (semitone). If a concert pitch frequency f_{A_4} (usually $440Hz$) corresponds to the midi-pitch number 69 then, the midi-pitch number of a frequency f, is

$$\nu = 69 + 12\log_2\Big(\frac{f}{f_{A_4}}\Big). \tag{3}$$

Taking the figure of the whole note as the unit, it is easy to define a note's duration coefficient $\delta \in \mathbb{R}$. A half note has a coefficient $1/2$, a quarter note $1/4$, a quaver $1/8$, etc, i. e.

$$\alpha = \frac{1}{2^a}, \quad -1 \leq a \leq 7. \tag{4}$$

In addition, each dotted note multiplies its duration by the factor

$$\beta = \sum_{k=0}^{b} \frac{1}{2^k} = \frac{2^{b+1} - 1}{2^b}, \tag{5}$$

where b is the number of dots of the note. On the other hand, tuplets (described in [2] as reading c notes in the space of d) are notated by the expression $c : d$, and modify the duration of each note with the factor

$$\gamma = \frac{d}{c}. \tag{6}$$

Taking into account expressions (4), (5) and (6), a number of τ tied notes will have a duration

$$\delta = \sum_{i=1}^{\tau}(\alpha_i \cdot \beta_i \cdot \gamma_i) = \sum_{i=1}^{\tau}\left[\frac{1}{2^{a_i}} \cdot \frac{2^{b_i+1} - 1}{2^{b_i}} \cdot \frac{d_i}{c_i}\right]. \tag{7}$$

Once the concept of musical note has been defined we can express a melody as an ordered sequence of n notes, being each note of the melody a point in a *metric space.*

Definition 1. *A melody is a sequence, $\mathscr{M} = \{x_i\}_{i=1}^{n}$, where each $x_i \in \mathbb{R}^q$ is a musical note.*

For instance, let us consider the melody in Figure 1. If we were only interested in the duration δ and in the midi-pitch number ν of each note, the fragment could be expressed as the following sequence of 14 notes:

$$\mathscr{M}_1 = \{(\delta_i, \nu_i)\}_{i=1}^{14} = \big\{(0.041667, 67), (0.041667, 69), (0.041667, 70), (0.166667, 69), (0.166667, 67), (0.166667, 70), (0.250000, 72), (0.041667, 69), (0.041667, 72) (0.041667, 70), (0.125000, 69), (0.187500, 67), (0.062500, 65), (0.750000, 67)\big\}.$$

If the melody is polyphonic, as in Figure 2, the notes' pitch is represented by a vector $\bar{\nu} \in \mathbb{R}^k$, where k is the least common multiple of the number of voices appearing in the melody. In Figure 2 notes with 1, 2 and 3 voices appear, then $k = \text{l.c.m.}(1, 2, 3) = 6$. Consequently, if our only interest are the duration and pitch of the notes, these could be expressed by using 7 coordinates. The corresponding melody would be $\mathscr{M}_2$.

$$\mathscr{M}_2 = \{(\delta_i, \bar{\nu}_i)\}_{i=1}^{5} = \big\{(0.125; 67, 67, 67, 71, 71, 71), (0.0625; 64, 64, 64, 67, 67, 67), (0.0625; 67, 67, 67, 71, 71, 71), (0.125; 64, 64, 67, 67, 71, 71), (0.125; 71, 71, 71, 71, 71, 71)\big\}.$$

Fig. 1. Example of a melodic line to be represented into the plane duration-pitch.

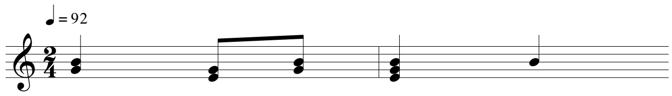

Fig. 2. Example of polyphonic melody.

3 Comparison of Melodies

If we consider two melodies $\mathscr{M}^A$ and $\mathscr{M}^B$, both belonging to a q-dimensional metric space, it is only possible to measure a well-defined distance between them if they have the same number of notes. This does not necessarily mean that both melodic lines have the same duration expressed in units of time; however, they need to have the same number of points.

In order to compare those melodies we will first calculate a total distance by accumulating the partial distance between each couple of notes $\mathbf{x_i}$, $\mathbf{y_i}, i \geq 1$, respecting the established order of the sequence of the points of both melodies.

Definition 2. *Let $\mathscr{M}^A = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ and $\mathscr{M}^B = \{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$ be two melodies with n notes, belonging to a q-dimensional space $\mathbb{R}^q$. Given $d : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}$ a distance function, the distance between $\mathscr{M}^A$ and $\mathscr{M}^B$ can be defined as*

$$D(\mathscr{M}^A, \mathscr{M}^B) = \mathscr{F}\{d(\mathbf{x_1}, \mathbf{y_1}), \ldots, d(\mathbf{x_n}, \mathbf{y_n})\}, \tag{8}$$

where $\mathscr{F}$ is a prefixed aggregation operator [15].

In this work, until the contrary is noticed, we will use as $\mathscr{F}$ operator the arithmetic mean, that is

$$\bar{D}(\mathscr{M}^A, \mathscr{M}^B) = \frac{1}{n}\sum_{i=1}^{n} d(\mathbf{x_i}, \mathbf{y_i}). \tag{9}$$

Nevertheless, regardless the operator chosen, it is easy to verify the following result:

Proposition 1. *Assuming the previous notation, the following inequalities hold*

$$\min_i d(\mathbf{x_i}, \mathbf{y_i}) \leq D(\mathscr{M}^A, \mathscr{M}^B) \leq \max_i d(\mathbf{x_i}, \mathbf{y_i}). \tag{10}$$

As well known [3], the Weber-Fechner Law approximates the psychological rules of human perception of intensity or pitch. This idea can be easily incorporated this into the calculation of the distance. For example, if we express the notes $\mathbf{x}$ with three coordinates (x_1, x_2, x_3) representing duration, pitch and intensity, respectively, in [9] the following distance is used

$$d(\mathbf{x}, \mathbf{y}) = \alpha \cdot |x_1 - y_1| + \beta \cdot |\log(x_2/y_2)| + \gamma \cdot |\log(x_3/y_3)|, \tag{11}$$

where α, β and γ are some prefixed constant values.

If we want to compare two melodies with different number of notes, Definition 2 has to be generalized. In fact, in the symbolic melody similarity literature it is possible to find several examples in which some definitions of distance between two melodies of different length are defined [10, 5]. The objective of many of these works is to approximate as much as possible to human perception [11]. With this aim, different techniques have been proposed ranging from the geometric structure of the melodies [1] to fuzzy logic [11], for instance.

A definition of an average distance based on the clustering of two melodies $\mathscr{M}^A = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ and $\mathscr{M}^B = \{\mathbf{y_1}, \ldots, \mathbf{y_m}\}$, being $n > m$, allows us to estimate how far away melody $\mathscr{M}^A$ is from melody $\mathscr{M}^B$. Despite the fact that this measurement will not satisfy the requirements of a distance function, the result provides some useful information about to the degree of similarity between these two melodies.

When a classical clustering process, e.g. *c-means clustering*, is applied to a general data set X of information, the result is a Boolean partition of X into c clusters, so each element of X belongs only to one cluster. Related to the comparison of melodies, we can use this procedure to cluster the set of n notes of melody $\mathscr{M}^A$ into m subsets. Once this is finished, we will be able to associate each subset in $\mathscr{M}^A$ to a note in $\mathscr{M}^B$ and finally, calculate an average distance from every point of each subset in $\mathscr{M}^A$ to its corresponding note in $\mathscr{M}^B$.

The global dissimilarity of the two melodies would be calculated by aggregating the partial average distance. However, while carrying out with this procedure we have to accept two arguable assertions:

1. It is assumed that comparing each note of $\mathscr{M}^A$ only to one note of $\mathscr{M}^B$ has musical sense.
2. In the process of comparing notes the order information is omitted. This is a key question in musical terms.

In what follows, a new proposal based on fuzzy logic will be presented. Real features of musical fact can be better represented by this new approach. With this objective, we will use fuzzy clustering applied to the calculation of a dissimilarity measure between two melodies of different number of notes.

3.1 Fuzzy c-means clustering (FCM)

The fuzzy c-means is a clustering method initially developed by J. C. Dunn [6] in 1973, based on the statement that any element of a given set is able to belong to more than one cluster. Thus, the fuzzy clustering method will provide a membership function that describes the belonging degree of each element to any centroid. As it is explained in [4], the generalization of fuzzy c-means algorithms comes from the iterative minimization of an objective functional.

Definition 3. *Let the data set* $X = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\} \subset \mathbb{R}^q$*. Let* v *be a set of cluster centers* $v = (\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_m})$*, with* $v_i \in \mathbb{R}^q$ *and* $m < n$*. Fuzzy c-means functionals are defined as*

$$J_\lambda = \sum_{i=1}^{n} \sum_{j=1}^{m} (u_{ij})^\lambda (d_{ij})^2, \tag{12}$$

where $d_{ij}^2 = \| \mathbf{x_i} - \mathbf{v_j} \|^2$*, being* $\| \cdot \|$ *any inner product induced norm on* $\mathbb{R}^q$*,* $\lambda \in [1, \infty)$ *is the weighting exponent (degree of fuzzyness of the process), and* u_{ij} *is the membership coefficient of* x_i *to the cluster* j*.*

The fuzzy clustering is achieved through an iterative optimization of J_λ, updating, at each iteration, the membership coefficients u_{ij} as well as the *cluster centers* v_j by using the following expressions

$$u_{ij} = \frac{1}{\sum\limits_{k=1}^{m} \left[\frac{\|x_i - v_j\|}{\|x_i - v_k\|} \right]^{\frac{2}{\lambda - 1}}}, \quad v_j = \frac{\sum\limits_{i=1}^{n} u_{ij}^\lambda \cdot x_i}{\sum\limits_{i=1}^{n} u_{ij}^\lambda} . \tag{13}$$

The matrix U is a fuzzy partition of X, formed by the membership coefficients u_{ij}

$$U_{ij} = \begin{pmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{nm} \end{pmatrix} . \tag{14}$$

As convergence condition of any fuzzy clustering we have

$$\sum_{j=1}^{m} u_{ij} = 1, \quad 1 \leq i \leq n. \tag{15}$$

3.2 Fuzzy c-means algorithm

In what follows we will show the implementation of the Fuzzy c-Means Clustering Algorithm proposed by Bezdek in [4].

FCM-Algorithm

STEP 1. Fix a number of clusters m, $2 \leq m < n$. Choose any inner product norm metric for $\mathbb{R}^q$; fix λ, $1 \leq \lambda < \infty$. Initialize $U^{(0)}$.

STEP 2. Calculate the fuzzy cluster centers $\{v_j^{(k)}\}$ with $U^{(k)}$ and expression (13).

STEP 3. Update $U^{(k)}$ using expression (13) and $\{v_j^{(k)}\}$.

STEP 4. Compare $U^{(k)}$ to $U^{(k+1)}$ using a convenient matrix norm, being $\epsilon \in (0,1)$ and arbitrary termination criterion. If $\| U^{(k+1)} - U^{(k)} \| \leq \epsilon$ then stop, otherwise set $k = k+1$ and return to STEP 2.

4 Measuring Dissimilarity by Means of Fuzzy Clusters

Let us consider two melodies $\mathscr{M}^A$ and $\mathscr{M}^B$ with different number of notes. We will now make a fuzzy partition of the notes from $\mathscr{M}^A$ with the initial cluster centers given by $\mathscr{M}^B$, and apply the FCM algorithm k times until the termination criterion is satisfied. Once the partition process is complete, we can define a dissimilarity function between $\mathscr{M}^A$ and $\mathscr{M}^B$ by using the final membership coefficients and the original cluster centers.

Definition 4. *Let $\mathscr{M}^A = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\} \subset \mathbb{R}^q$ and $\mathscr{M}^B = \{\mathbf{y_1}, \ldots, \mathbf{y_m}\} \subset \mathbb{R}^q$ be two melodies, where $n > m$. Let $d : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}$ be a distance function. Let u_{ij} be the final membership coefficients calculated with FCM algorithm. The average dissimilarity $\mathscr{D}$ from $\mathscr{M}^A$ to $\mathscr{M}^B$ is defined by*

$$\mathscr{D}(\mathscr{M}^A, \mathscr{M}^B) = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} u_{ij} \cdot d(\mathbf{x_i}, \mathbf{y_j}) . \tag{16}$$

By construction, $\mathscr{D}$ does not consider the natural order of the sequence of notes within each melody. Thus, the partition that FCM algorithms calculate does not weight in any special way the notes whose degree of neighbourhood is stronger. As an illustrative example of this fact, in Figure 3 it is possible to see three different melodies. Since Melody B is a complete retrogradation of Melody A, average dissimilarity $\mathscr{D}$ between melodies A and C has exactly the same value than average dissimilarity between melodies B and C,

$$\mathscr{D}(\mathscr{M}^A, \mathscr{M}^c) = 0.23354, \qquad \mathscr{D}(\mathscr{M}^B, \mathscr{M}^c) = 0.23354.$$

This example shows that a comparison of different melodies without taking into account the order of the notes does not completely reflect musical reallity. To avoid this, we will introduce a dependence with the order in the algorithm.

Fig. 3. Three example melodies, where Melody B is a retrogradation of Melody A.

In this way, higher weights will be given to the pair of notes that share closer positions in the order of each melody, reducing the contribution to the global dissimilarity of the pair of notes that are far away from an ordinal point of view. Neighbourhood functions will provide the information related to the order in which the pair of notes must be compared.

Definition 5. *A continuous function* $f : \mathbb{R}^2 \to \mathbb{R}$ *is a neighbourhood function between two melodies* $\mathscr{M}^A, \mathscr{M}^B$ *if*

$$\int_1^n f(i,j)di < \infty, \quad \forall j \in \{1,2,\dots,m\}, \tag{17}$$

where n *is the number of notes of* $\mathscr{M}^A$ *and* m *the number of notes of* $\mathscr{M}^B$*.*

If a correct setting for the neighbourhood function is defined, neighbourhood values of $i \in (j-\varepsilon, j+\varepsilon)$ will be assigned to higher coefficients and the rest of values will be assigned lower coefficients.

The procedure will be following: Once the fuzzy partition U has been calculated, we will assign a weight to any element u_{ij} by means of a specific neighbourhood function $f(i,j)$. In order to accomplish with the FCM convergence criterion, we will normalize U as follows

$$\tilde{u}_{ij} = \frac{u_{ij} \cdot f(i,j)}{\sum\limits_{k=1}^{m} u_{ik} \cdot f(i,k)}. \tag{18}$$

Example 1. Gaussian neighbourhood function

$$f_G(i,j) = Ae^{-\frac{1}{2\sigma^2}\left[i+1-\frac{(n-1)\cdot(j-1)}{(m-1)}\right]^2}. \tag{19}$$

In this function it is easy to see how the original μ value has been replaced by $1 - \frac{(n-1)\cdot(j-1)}{(m-1)}$. This expression has been obtained from the equation of a line $\mu = f(j)$ that crosses through points $(1,1)$ and (m,n). Given the fixed values $n, m \in \mathbb{N}$, the shape of $f(i,j)$ will change for each pair of values i, j. When $j = 1$, the Gaussian will be centered at $i = 1$, but when $j = m$ it will be centered on $i = n$.

Our proposal is to modify the algorithm FCM in such a way that the order of the sequences of the notes $\mathscr{M}^A = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\}$ and $\mathscr{M}^B = \{\mathbf{y_1}, \ldots, \mathbf{y_m}\}$, $n < m$, is taken into account. With this objective we propose the following algorithm, named fuzzy ordered c-means (FOCM).

4.1 FOCM-Algorithm

- STEP 1. Set $\{v_j^{(0)}\} = \{y_j\}$. Let m, n be the number of notes of $\mathscr{M}^B$ and $\mathscr{M}^A$, respectively. Choose any convenient neighbourhood function.
- STEP 2. Choose any inner product norm metric for $\mathbb{R}^q$, and fix $\lambda \geq 1$. Calculate the initial $\widetilde{U}^{(0)}$ using (13), (18) and $\{v_j^{(0)}\}$.
- STEP 3. Calculate the fuzzy cluster centers $\{v_j^{(k)}\}$ with $\widetilde{U}^{(k)}$ and the equation (13).
- STEP 4. Update $\widetilde{U}^{(k)}$ using the equations (13), (18) and $\{v_j^{(k)}\}$.
- STEP 5. Compare $\widetilde{U}^{(k)}$ to $\widetilde{U}^{(k+1)}$ using a convenient matrix norm; being $\epsilon \in (0,1)$ and arbitrary termination criterion. If $\| \widetilde{U}^{(k+1)} - \widetilde{U}^{(k)} \| \leq \epsilon$ then stop; otherwise set $k = k + 1$ and return to STEP 3.

Once the melodies have been compared by taking into account all the described characteristics, we can establish the following definition.

Definition 6. *Let $\mathscr{M}^A = \{\mathbf{x_1}, \ldots, \mathbf{x_n}\} \in \mathbb{R}^q$ and $\mathscr{M}^B = \{\mathbf{y_1}, \ldots, \mathbf{y_m}\} \in \mathbb{R}^q$ be two melodies of different number of notes. Let $d : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}$ be a distance function. Let $\tilde{u}_{ij}$ be the final membership coefficients calculated with FOCM algorithm. The average ordered dissimilarity $\widetilde{\mathscr{D}}$ from $\mathscr{M}^A$ to $\mathscr{M}^B$ is defined by*

$$\widetilde{\mathscr{D}}(\mathscr{M}^A, \mathscr{M}^B) = \frac{1}{n \cdot m} \sum_{i=1}^{n} \sum_{j=1}^{m} \tilde{u}_{ij} \cdot d(\mathbf{x_i}, \mathbf{y_j}). \tag{20}$$

In what follows we show the utility of expression (20). For this, we will calculate the dissimilarity between different melodies.

4.2 Computational Examples

Example 2. We are now going to compare the melodies appearing in Figure 4 using expressions (16) and (20).

The dissimilarity values are $\mathscr{D}(\mathscr{M}^A, \mathscr{M}^B) = 0.68938$ and $\widetilde{\mathscr{D}}(\mathscr{M}^A, \mathscr{M}^B) = 3.93483$. The reason behind the disparity in the obtained results is that when the order of the notes is not taken into account, we use $\mathscr{D}$, sharp notes in $\mathscr{M}^A$ are compared with sharp notes in $\mathscr{M}^B$ and flat notes in $\mathscr{M}^A$ are compared with flat notes in $\mathscr{M}^B$. In fact, when we give importance to the order, $\widetilde{\mathscr{D}}$, both melodies are quite different (dissimilarity is almost six times greater with $\widetilde{\mathscr{D}}$ than with $\mathscr{D}$). In Figure 5 we show a screenshot appearing in our implementation of algorithm FOCM. We can observe how, for example, the two first notes in $\mathscr{M}^B$ are associated to the first time measure in $\mathscr{M}^A$, showing the above mentioned differences.

Fig. 4. Melodies of Example 1.

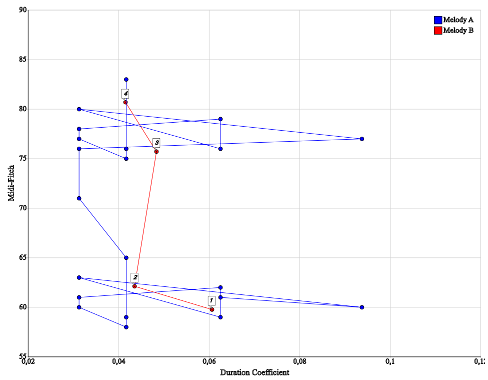

Fig. 5. Final result of clustering algorithm FOCM of $\mathscr{M}^A$ and $\mathscr{M}^B$.

Fig. 6. Melodies of Example 2.

Example 3. Using the melodies displayed in Figure 6, we will now show the capacity of our proposal to measure the dissimilarity in polyphonic melodies.

The obtained values are $\widetilde{\mathscr{D}}(\mathscr{M}^A, \mathscr{M}^C) = 0.30695, \widetilde{\mathscr{D}}(\mathscr{M}^B, \mathscr{M}^C) = 0.46356$. As it was expected, melodies in this example are more similar than melodies in Example 1 and the differences between them increase when one of them is polyphonic and the other is not.

Example 4. In Table 1 we provide an example showing how our proposal functions. We have selected four passages of very well-known musical works: (1) = W. A. Mozart. Symphony No. 40. First movement. Measures 1-4, (2) = L. V.

Beethoven. Symphony No. 6. First movement. Measures 1-4, (3) = J. Brahms. Symphony No. 3. Second movement. Measures 1-8, and (4) = B. Bartók. Music for strings, percussion and celesta. First movement. Measures 1-4.

Table 1. Measurement of dissimilarity between melodies from Example 4.

$\mathscr{M}^A$	$\mathscr{M}^B$	$\widetilde{\mathscr{D}}$	♯States	$\widetilde{\mathscr{D}}$	♯States	$\widetilde{\mathscr{D}}$	♯States
		Fuzzy coeficient = 2		Fuzzy coeficient = 2.5		Fuzzy coeficient = 3	
1	2	0.2395874272	389	0.1820221001	389	0.1835055288	17
3	1	0.1144880336	56	0.1079920868	375	0.1094903790	584
4	1	0.4231256333	508	0.3294539142	672	0.3609358015	636
3	2	0.0910976980	10	0.1231091704	20	0.1233827179	12
4	2	0.7071511697	81	0.6642747436	194	0.6666523284	353
3	4	0.0692844111	51	0.1451223554	429	0.1383288250	310

In Table 1 we display the dissimilarity measures between melodies from Example 4, as well as the number of intermediate compositions (♯States) generated by the algorithm with different values of the fuzzy coefficient used in the FOCM.

Fig. 7. One of the intermediate melodies obtained by the algorithm when measuring the dissimilarity between passages (1) and (3) from Example 4.

5 Conclusions

The evaluation of the degree of likeness of two melodies is nowadays a topic of great interest. By comparing the similarity of different melodies it is possible to find patterns, to extract rules and to identify structures, all key questions in the study of musical styles. In this work, we have proposed a fuzzy logic tool, fuzzy c-means clustering, for the measurement of the similarity between two melodies with different number of notes.

The proposed FOCM algorithm allow us to define a measurement of the symbolic melodic dissimilarity between two different melodies, taking into account the order of the sequences of notes that each melody contains. To a certain extend, the definition of *fuzzy c-means average ordered dissimilarity* offers a geometric way to compare very different melodic lines that can be used with several purposes, like classification of melodies, computer-aided composition or musical-styles recognition.

Our proposal could also be applied to other fields of research in which is necessary to estimate the degree of closeness of two different sequences of ordered information.

References

1. Aloupis, G., Fevens, T., Langerman, S., Matsui, T., Mesa, A., Nunez, Y., Rappaport, D., and Toussaint, G.: Algorithms for computing geometric measures of melodic similarity. Computer Music Journal, 30 (3), 67–76 (2006)
2. Apel, W.: Harvard Dictionary of Music: Second Edition.The Belknap Press of Harvard University Press, Cambridge, Massachussets (1994)
3. Benson, D.: Music: a Mathematical Offering. Cambridge University Press, Cambridge (2006)
4. Bezdek, J. C.: Pattern Recognition with Fuzzy Objective Function Algoritms. Plenum Press, New York (1981)
5. Downie, J. S.: Evaluating a Simple Approach to Musical Information retreival: Conceiving Melodic N-grams as Text. (PhD Thesis). University of Western Ontario, Ontario (1999)
6. Dunn, J. C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics, 3, 32–57 (1973)
7. Haluska, J.: The Mathematical Theory of Tone Systems, Marcel Dekker, Inc., Bratislava (2005)
8. Liern, V.: Fuzzy tuning systems: the mathematics of the musicians. Fuzzy Sets and Systems 150, 35–52 (2005)
9. Liern, V.: La música y sus materiales: una ayuda para las clases de matemáticas. Suma, 14: 60–64 (1994)
10. Mongeau, M., and Sankoff, D.: Comparison of Musical Sequences. Computers and the Humanities, 24, 161–175 (1990)
11. Müllensiefen, D., and Frieler, K: Cognitive adequacy in the measurement of melodic similarity: Algorithmic vs. human judgments. Computing in Musicology, 13, 147–176 (2004)
12. Selfridge-Field, E.: Beyond MIDI: the handbook of musical codes, MIT Press, Cambridge, Massachussets (1997)
13. Velardo, V., Vallati, M., and Jan, S.: Symbolic Melodic Similarity: State of the Art and Future Challenges, Computer Music Journal, 40 (2), 70 (2016)
14. Xenakis, I.: Formalized music: thought and mathematics in composition, Pendragon Press, Launceston (1992)
15. Yager, R. R.: On ordered weighted averaging aggregation operators in multi-criteria decision making. IEEE Transactions on Systems, Man and Cybernetics 18, 183–190 (1988)
16. Zadeh, L. A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)