Similarity Measures for Rhythmic Sequences

João M. Martins [1] , Marcelo Gimenes [1] , Jônatas Manzolli [2] , Adolfo Maia Jr. [2]

[1]Computer Music Research, Faculty of Technology
University of Plymouth, Plymouth, Devon PL4 8AA
United Kingdom.

[2]Departamento de Matemática Aplicada, IMECC and
Núcleo Interdisciplinar de Comunicação Sonora (NICS), UNICAMP,
13.081-970, Campinas, SP, Brazil.

joao.martins@plymouth.ac.uk, marcelo.gimenes@plymouth.ac.uk

jonatas@nics.unicamp.br, adolfo@nics.unicamp.br

Abstract. *This paper presents a new model for measuring similarity in a general Rhythm Space. Similarity is measured by establishing a comparison between subsequences of a given rhythm. We introduce the hierarchical subdivision of rhythm sequences in several levels, and compute a Distance Matrix for each level using the "block distance". The information about the similarity of the rhythmic substructures is retrieved from the matrices and coded into a Similarity Coefficient Vector (SCV). We also present possibilities for the reduction to single values of similarity derived from the SCV. In addition, two applications of the formal model are presented, showing the potential for development using this approach.*

1. Introduction

"To study rhythm is to study all of music. Rhythm both organises, and is itself organised by, all the elements which create and shape musical processes" and this is how Meyer and Cooper [Cooper G., 1963] emphasize the importance of rhythm in the overall structure of music. In the 20^{th} century, rhythm in music, has finally been put in the focus of the attention of composers, such as Igor Stravinsky, Olivier Messiaen, Iannis Xenakis. Since then, an increasing interest to automatically compare music styles and composing techniques. Rhythm, as the most fundamental aspect of music plays a decisive role in this task.

The rhythm organization of music, as well as speech sounds and environmental events, are highly dependent on the perception of human beings. Even when subjects are presented with equal pulses at equally spaced intervals, the pulses are perceived as being grouped in a regular metric structure [Handel, 1989]. It is argued that this implicit metrical organization improves attention and memorisation of sequential tasks.

Perception influenced a great number of researchers in music [Gabrielsson, 1973, Povel and Essens, 1985]. Cooper and Meyer [Cooper G., 1963], developed an auditory theory based on *Gestalt* theories of perception, where rhythm groups in the basic level are seen as units that are categorised according to the position of the accentuated notes. Also according to them, one strong cue in rhythm organization is the one of pattern repetition. When a rhythmic motive is repeated, the brain integrates it creating a unit that is memorised and categorised accordingly.

In our approach we provide a measure of the occurrence of these repetitive patterns in a stream of rhythmic events. We leave out of this study the implications of accentuation, melody, harmony, timbre, and articulation to the perception of rhythm, and we do so for two reasons: Firstly, we are able to extract interesting and meaningful information solely from the position were the events take place, and secondly, we can find repertoire for percussion that does not contemplate any of the former musical characteristics apart from accentuation. We strongly believe, though, that our measure can be extended to incorporate some of these characteristics. Furthermore, we can have rhythms which do not obey the marks of bars or any metrical structure. This enables to compare and distinguish rhythmic sequences with different subdivisions, and possibly to provide some insight on situations that metric is difficult to extract.

Computers find it simple to discriminate if something is equal or different, but the problem rises when there is the need to evaluate if something is similar [Minsky, 1988]. The necessity of similarity measures concerns many areas of music research, specially music information retrieval systems [Hewlett and Selfridge-Field, 2005], automatic rhythm transcription of human-performed music to MIDI protocol [Takeda et al., 2003], evaluation of copyright issues, and evolutionary music [Miranda, 2004].

On the side of the abstract models, interesting results were achieved using the Levenshtein distance, also called edit distance. This a popular method for measuring similarity between strings of text of arbitrary length. This method counts the number of insertions, deletions and substitutions necessary to change one string into another other, being this number the measure of similarity between the sequences. Orpen and Huron have applied this distance to measure melodic, rhythmic and harmonic similarity in Bach chorales [Orpen and Huron, 1992]. Mongueau and Sankoff provided a method which can be seen as an extension of the previous [Mongeau and Sankoff, 1990]. Instead of considering that each transformation to the sequence contributes with the value of one to the distance, each transformation contributes with a weighted value sensitive to the kind of musical differences who are to be measured.

In this work we are most interested in constructing an abstract and formal model which can be able to compare rhythm patterns in the most general way, capturing information in several layers of detail. In addition we intend our model to be able to manipulate rhythm sequences in order to create new ones, which could be used in music composition. In the future we will extend this work by comparing it with the existing formal models and we hope to establish a closer relation between our model and perception by testing rhythmic similarity with human subjects.

In the next section we formally introduce the concepts of Rhythm Space and Similarity Measure. In section 3 we describe our algorithm implementation. In section 4 we present two applications of our model. In the last section we conclude with some comments about the model and list some interesting topics for further research.

2. Rhythm Space and Similarity Measure

In this work, rhythm sequences are thought as elements (or vectors) of a finite dimension vector space. Formally we have coded rhythms as sequences of numbers $(b_1, b_2, \ldots, b_r,)$, where the entries b_i can be any number of the set $B = \{-1, 0, 1, 2, \ldots, J\}$ which we named *Beat Set*. A positive number in a sequence indicates first that one have a beat and its magnitude indicates the level of acentuation, such as strong beat, weak beat, half strong beat, etc. The number of sequencial 0s indicates the duration of the beat. The number -1 indicates pauses or, in MIDI protocol, a note off. We associate the positive numbers to accentuation. For example, taking $J = 3$ we get the Beat Set $B = \{-1, 0, 1, 2, 3\}$, the accentuation should be read as: 1 means a *weak beat*, 2 a *half strong* and 3 a *strong beat*. Then we can construct rhythm sequences like, for example, $(3, 0, 1, 0, -1, 0, 0, 0, 2, 0, 0, 0, 3)$. Clearly we can have as many accentuation we wish, just extending the Beat Set. However we must introduce a prescription in order to avoid some ambiguities.

Rule: *If in a rhythm sequence a value -1 occurs, it can only occur again after a positive number had occurred first.*

This rule avoids ambiguities, for example, if we compare the sequences like $a = (1, -1, -1, 0, 0)$ and $b = (1, -1, 0, 0, 0)$. Since they are different one from another, the distance (see below) between them is positive. Nevertheless they represent the same events (in this case, pause) which intuitively suggests the distance must be zero. Our rule says that only the second sequence b is a valid one, that is, it is an element of our Rhythm Space *R* defined below.

Now, given a Beat Set *B*, we define its associated n-rhythm vector space $R_n(B)$ as the set of all n-vectors $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ in which each entry is an element of the Beat Set *B*. On $R_n(B)$ we can define a distance. Let $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_n)$ be two vectors in $R_n(B)$. The p-distance between them is defined as

$$d_p(\mathbf{v}, \mathbf{w}) = \left(\sum_{i=1}^{n} |v_i - w_i|^p \right)^{1/p}. \tag{1}$$

The value of p can be chosen according to the application or the kind of music considered. In our examples and applications we take, for the sake of simplicity, $p = 1$, the so called *block distance*.

The above p-distance is defined only for rhythm sequences which have the same length. Obviously, in most of applications, we must compare rhythm sequences of different lengths. Although it is possible to define a distance between vectors with different sizes (the so called Hausdorff Distance) we prefer to use for comparison of arbitrary rhythm sequences the concept of similarity in a particular way. So, the next logical step is to put together all the possible rhythm sequences into a same *Rhythm Space* and define a *Similarity Measure* on it. This is as follows.

Firstly, we define the **Rhythm Space**, denoted here by *R*, as the union of all $R_n(B)$, that is, $R = \bigcup_{i=0}^{\infty} R_n(B)$. We name, alternatively, the elements of *R* as *Rhythm Vectors*. Note that for each given Beat Set we have an associated Rhythm Space. We introduce similarity measure on *R* as follows. Given an arbitrary rhythm vector $\mathbf{v} = (v_1, v_2, \ldots, v_n)$, we define a k-level subsequence $\mathbf{v}^{(k)}$ of $\mathbf{v}$ as any subsequence with k elements extracted from $\mathbf{v}$, preserving the original order of $\mathbf{v}$. For example, if $\mathbf{v} = (2, 0, 0, 1, -1, 0, 1, 1)$ we can extract five ordered four-levels sequences, namely, $\{(2, 0, 0, 1), (0, 0, 1, -1), (0, 1, -1, 0), (1, -1, 0, 1), (-1, 0, 1, 1)\}$. It is easy to see that a vector with n elements has $n - k + 1$ k-level subsequences. Now, given two rhythm vectors $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ and $\mathbf{w} = (w_1, w_2, \ldots, w_m)$ in *R* consider all k-level sequences of both vectors, that is, the sets $S_{\mathbf{v}}^{(k)} = \{\mathbf{v}_i^{(k)}, i = 1, 2, \ldots, n - k + 1)\}$ and $S_{\mathbf{w}}^{(k)} = \{\mathbf{w}_j^{(k)}, j = 1, 2, \ldots, m - k + 1)\}$. If, for example, $m \leq n$ we only can consider sequences with length smaller than m, that is, we must take $1 \leq k \leq min(m, n)$.

Formally, for each k-level, we define the (i, j)-elements of the *k-level Distance Matrix* $\mathbf{D}^{(k)}$of two vectors $\mathbf{v}$ and $\mathbf{w}$ as:

$$\left[\mathbf{D}^{(k)}(\mathbf{v}, \mathbf{w})\right](i, j) = d_p(\mathbf{v}_i^{(k)}, \mathbf{w}_j^{(k)}) \tag{2}$$

where $i = 1, 2, \ldots, n-k+1$ and $j = 1, 2, \ldots, m-k+1$. Below we show a visualization of a general k-level Distance Matrix.

$$\mathbf{D}^{(k)} = \begin{bmatrix} d_p(\mathbf{v}_1^{(k)}, \mathbf{w}_1^{(k)}) & d_p(\mathbf{v}_1^{(k)}, \mathbf{w}_2^{(k)}) & \dots & d_p(\mathbf{v}_1^{(k)}, \mathbf{w}_{(m-k+1)}^{(k)}) \\ d_p(\mathbf{v}_2^{(k)}, \mathbf{w}_1^{(k)}) & d_p(\mathbf{v}_2^{(k)}, \mathbf{w}_2^{(k)}) & \dots & d_p(\mathbf{v}_2^{(k)}, \mathbf{w}_{(m-k+1)}^{(k)}) \\ \vdots & \vdots & \vdots & \vdots \\ d_p(\mathbf{v}_{(n-k+1)}^{(k)}, \mathbf{w}_1^{(k)}) & d_p(\mathbf{v}_{(n-k+1)}^{(k)}, \mathbf{w}_2^{(k)}) & \dots & d_p(\mathbf{v}_{(n-k+1)}^{(k)}, \mathbf{w}_{(m-k+1)}^{(k)}) \end{bmatrix}$$

Although there exist many different measures we restrict our analysis, as mentioned above, to the block distance ($p = 1$) and the Beat Set to $\mathbf{B} = \{0, 1\}$ (Fig. 1).

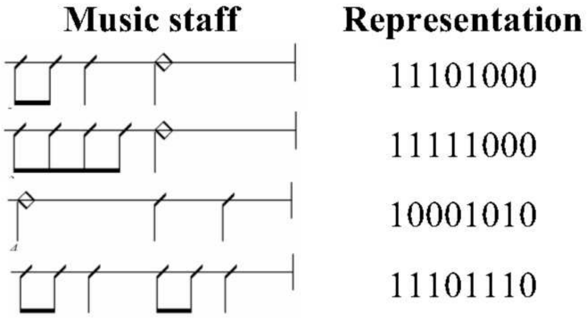

Figure 1: Musical notation and correspondent coding

We get, then, a k-level Distance Matrix whose elements are non negative integers. Now, we define the *k-level Similarity Coefficient* as the

$$c^{(k)}(\mathbf{v}, \mathbf{w}) = \frac{z(k)}{(n-k+1)(m-k+1)} \tag{3}$$

where $z(k)$ is the number of zeros in the matrix $\mathbf{D}^{(k)}$. Roughly speaking the similarity coefficient measures the *sparsity* of the matrix $\mathbf{D}^{(k)}$. Greater the coefficient $c^{(k)}$, greater is the similarity between the subsequences of level k. In the extreme case a matrix with all coefficients equal to 1, it means that one of the sequences has a perfect copy of it contained in other one.

Now we can collect all the k-levels coefficients in a vector we name *Similarity Coefficient Vector* (SCV). It reads like

$$\mathbf{C} = \left[c^{(1)}, c^{(2)}, \dots, c^{min(m,n))}\right] \tag{4}$$

In Fig. 2 we show an example of the 3-level Distance Matrix and its respective SCV. Bellow we provide an example of this approach.

Example:
Take the Beat Set as $B = \{0, 1\}$. Let $\mathbf{v} = (1, 0, 1, 1)$ and $\mathbf{w} = (1, 0, 1, 1, 0, 1)$ be two rhythm sequences in *R*. The possible 3-level sequences for **v** and **w** are:

- $S_{\mathbf{v}}^{(3)} = \{(1, 0, 1), (0, 1, 1)\}$
- $S_{\mathbf{w}}^{(3)} = \{(1, 0, 1), (0, 1, 1), (1, 1, 0), (1, 0, 1)\}$

Let us take, for the sake of simplicity, $p = 1$ on the Rhythm Space. Since we must take the distance between all elements of each level up to sixth level, one can guess that a large number of evaluations is needed. We show below only the distance between the

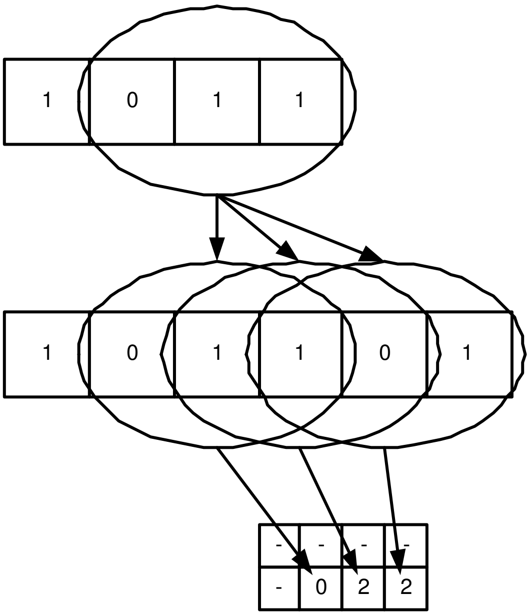

Figure 2: Building the Distances Matrix for the 3-level

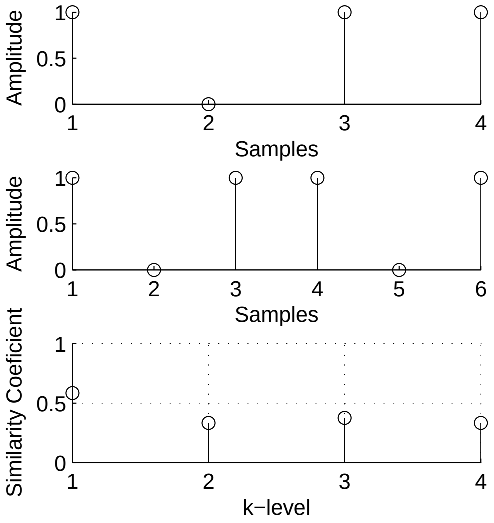

Figure 3: Example of two vectors and their Similarity Coefficients Vector (SCV)

combinations in the 3-level. According to the procedure describe above and shown in Fig. 2, we obtain the (2×4) 3-level Distance Matrix:

$$D^{(3)}(\mathbf{v}, \mathbf{w}) = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 2 & 0 & 2 & 2 \end{bmatrix} \tag{5}$$

The 3-level coefficient can be read easily from this matrix and it is $c^{(3)} = 3/8 = 0.375$.

The complete SCV (Fig. 3) for the above example is given by

$$\mathbf{C} = [0.5833, 0.3333, 0.3750, 0.3333] \tag{6}$$

In the next section we show the algorithmic implementation and make some additional comments on our model.

3. Algorithm Implementation

We have implemented an algorithm in MATLAB which is able to construct, manipulate, and play the rhythm sequences defined by our model. We restrict our analysis below to rhythms without accentuation and articulation and also without pauses. This is a crude approximation to real rhythms and, in our model, it is accomplished by taking the simplest Beat Set, that is, $B = \{0, 1\}$. These aspects will be added in a further implementation of our formal modal described above.

We also devised a function to play back the input sequences, to do a subjective evaluation of the result of the measure. The events correspond to sinusoidal functions with exponential decay and we introduced a short tone at the starting point as a reference for the beginning of the sequence.

3.1. Similarity Coefficient Vector

The algorithm picks two sequences of elements extracted from the Beat Set and computes, for each k-level, the matrix $\mathbf{D}^{(k)}$. The meaning of a zero in a matrix element, corresponds to a perfect match between sub-sequences of the two input vectors. At this point we have as many matrices D as the length of the shortest input vector.

The sparsity of the matrix, which means the number of zeros in each $\mathbf{D}^{(k)}$ matrix, will give information on how similar are the subsequences of that particular k-level. The algorithm computes the ratio between the number of zeros and the product of the dimensions of $\mathbf{D}^{(k)}$ for each k-level and stores those values in the SCV.

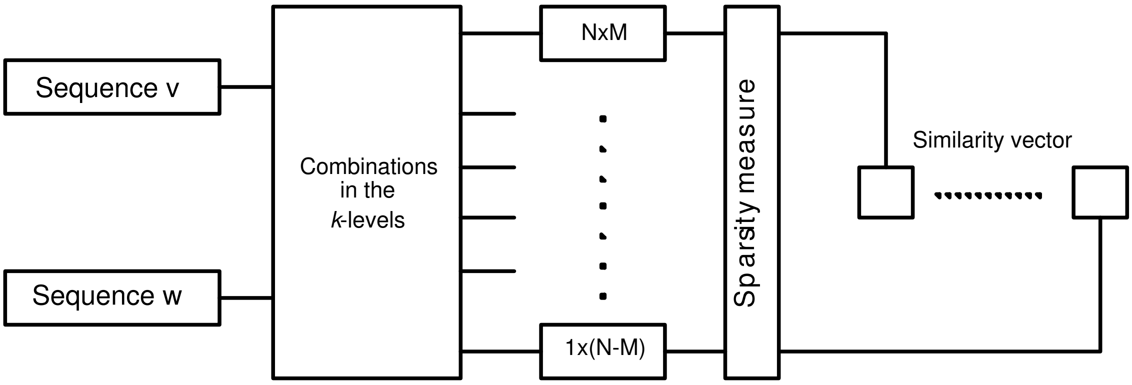

Figure 4: Model for the creation of the SCV between the two vectors

3.2. Analysis by Single Values

In addition to the SCV, we thought that it would also be interesting to reduce the quest for measuring similarity to a single value, enabling an easier comparison between the sequences. We offer three different solutions for this problem.

It is clear to us that the content of the first element provides a low level of information, as it only tells us wether the distribution of events (1s) and no-events (0s) in both of the vectors is even, or it is polarized towards having more events or no-events. On the other hand, the content of the last element of the SCV gives us the highest information. Finding a non-zero value in this position implies that the shortest input sequence exists at least once in the longest input sequence. In most of the comparisons this element will be zero. So the last non-zero element will tell us that what is the size of the longest sub-sequence that is common to both input sequences.

Another parameter that may be useful is the sum of the elements of SCV, which will take into account the coefficients from all the k-levels. For example, by considering

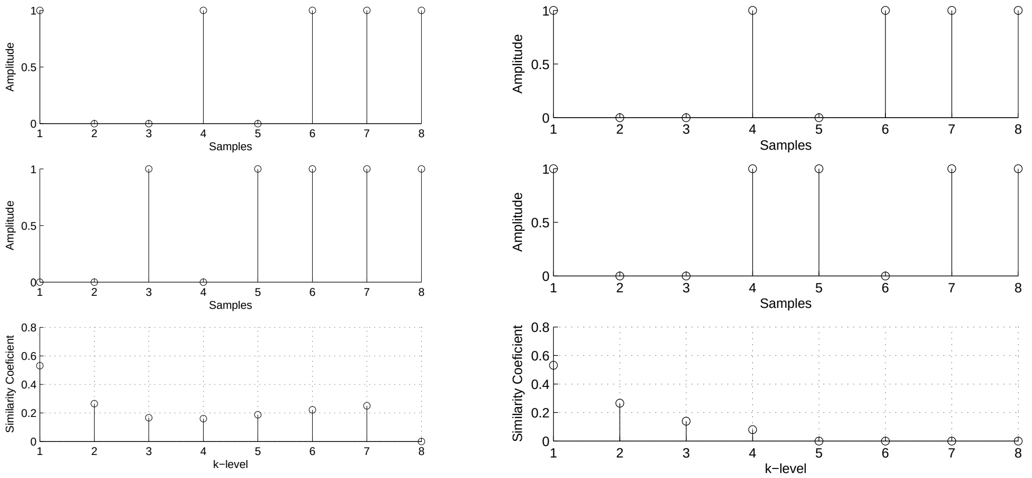

Figure 5: Vector with high values of the Similarity Coefficients values

the input sequence $\mathbf{v} = (1, 0, 0, 1, 0, 1, 1, 1)$ we ran the distance for all possible **w** vectors of length 8. In Fig. 5 we present two vectors that show high similarity with the presented sequence. By maximizing the sum of the Similarity Coefficients vector and removing the input vector from the competition, we arrive to the value $\sum C_i = 1.7829$ with the most similar vector being $\mathbf{w} = (0, 0, 1, 0, 1, 1, 1, 1)$ as can be seen in Fig. 5 (left). If instead we use the same vector and minimize the sum of the SCV elements, we get the least similar vector. For the example above the resultant value is $\sum C_i = 0.5179$, and the resultant **w** vector will be the no-event vector.

However, the sum above, does not consider, that there is greater importance in the rightmost elements of the vector. This can be achieved by taking a weighted sum of the elements, with an increasing profile of the weights.

4. Computer Applications

The formal model presented in this work is flexible enough to be used in several applications from rhythm analysis to creation of new rhythms, etc. Below we describe just two applications, namely, *Net-Rhythms* related to Neural Networks, developed by one of the authors(JM), and *RGeme*developed by another author (MG) in which AI agents learn rhythmic sequences from one another.

4.1. Neural Networks and Rhythms

Net-rhythms is a tool developed to classify and store rhythmic representations in a neural network. The framework used by this tool is constituted by a Neural Network called the SARDNET [James and Miikkulainen, 1995], an extended Kohonen self-organising feature map [Kohonen, 1985]. This network was developed to study the study of sequences and organization of phonemes in the context of language. We decided to explore its potential in the representation of rhythmic sequences, and new problems arose particularly related to the measurement of the distance between two vectors. The diagram on Fig. 6 explains how the network works.

The rhythms are coded according to the representation depicted in Fig. 1. Whenever a small rhythmic sequence $\mathbf{v}_t$ in time step t reaches the input, the distance from that sequence to all weight vectors w_j is computed. The neuron corresponding to weight more similar to the input, according to the defined distance, is activated and removed from further testing. As time progresses all activations are decayed, implying that after some time

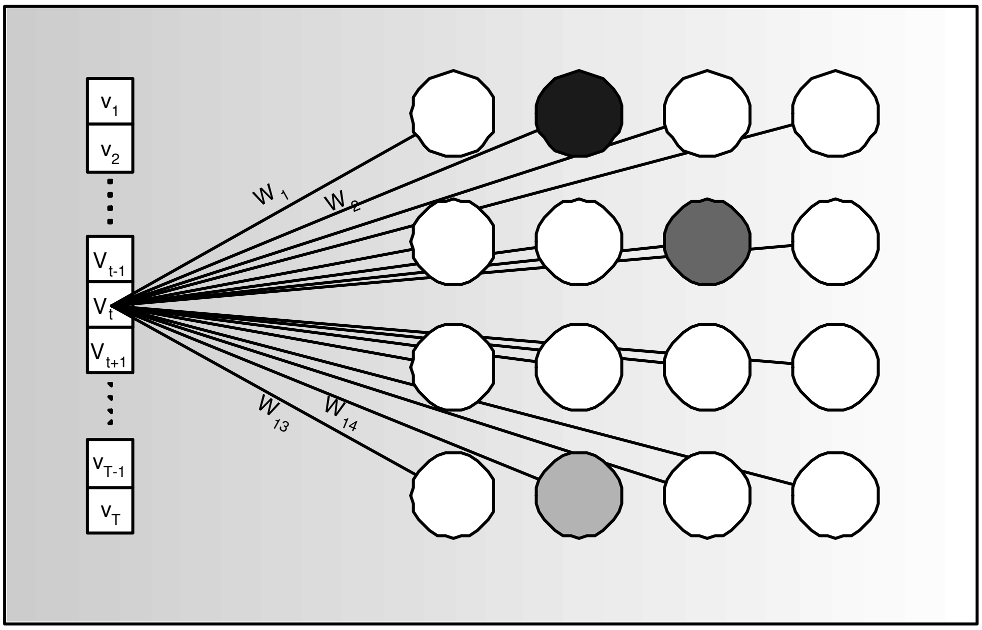

Figure 6: Diagram of the Sardnet with three activated neurons

steps there will be a ladder of activations in the network. In Fig. 6 there are three activated neurons represented by grey tonalities, corresponding the $\mathbf{w}_{14}$ weight to the first activated neuron, and $\mathbf{w}_2$ to the last one. Finally, after a complete sequence on time T, the weights from the activated neurons are slightly adapted in order to decrease their distances to input vectors.

This network can represent in a bidimensional space the rhythms that arrive sequentially to the input, and self-organize simulating a learning procedure.

As stated before, the choice of the wining neuron implies the measurement of the distance from the input to each of the neurons from the Sardnet. The distance proposed by the original creators of the network was the Euclidean distance, however there are some problems as this measure. The Euclidean distance does not allow sequences with different lengths and does not capture the similarity between equal sub-sequences that have their position shifted in time. The use of the SCV, and the other measures presented in Sec. 3.2, help solving the problems presented above.

4.2. Agents and Memes: A Rhythm Imitation Society

RGeme is an artificial intelligence system for the composition of rhythmic passages inspired by Richard Dawkin's theory of memes that is being presently developed in the Future Music Lab at the University of Plymouth. According to Dawkins [Dawkins, 1989, Dawkins, 1991], memes are basic units of cultural transmission in the same way that genes, in biology, are units of genetic information. Other researchers have already studied some applications of this concept in music [Cox, 2001, Gabora, 1997]

In *RGeme* a rhythmic composition is understood as the process of interconnecting sequences of basic elements (or ”rhythmic memes”) that have varied roles in the stream. Intelligent agents learn these roles from examples of musical pieces in order to evolve a ”musical worldview” which consists of a ”style matrix” of basic rhythms. During the learning stage, agents parse examples of pieces of music in search for rhythmic memes. These (candidates) memes are then compared with the agent's database of memes which we named Style Matrix (see Table 1), and are stored or transformed accordingly. For example, if the candidate meme is not already present in the agent's Style Matrix, it is copied and it's weight is set to 1. Now the model of distance of rhythm patterns is used in this application in order to upgrade the weight of each meme in Style Matrix. In this way the style of the memes evolves in time. In *RGeme* it was used the block distance. So all memes in the agent's style matrix have their weight upgraded according to their distance to the candidate meme.

Meme	dFL	dLL	nL	W
01011101	1	1	6	1.0385
11011101	1	1	31	1.0424
10001000	1	1	1	1.0181
10010101	1	1	1	1.0171
11011010	1	1	1	1.0159
10011010	1	1	4	1.0090
10011001	1	1	4	1.0075
11111111	1	1	1	1.0040
10000000	1	1	1	1.0000

Table 1: Extract from an Agent Style Matrix

where

- dFL: date the meme was first listened to
- dLL: date the meme was last listened to
- nL: number of times the meme was listened to
- W: upgraded weight

In the second stage, the system creates new rhythmic sequences (Production Phase) according to the musical structures and rules that were previously extracted from the styles of the pieces that were used in the learning stage. At this stage, agents are able to learn from each other's ”compositions” and capable of evolving new rhythmic styles by adapting to each other's rhythms. Clearly, new distances and similarity measures as shown above can be implemented in *RGeme*, which, of course could result in a different evolution of the memes society. This is presently under investigation.

5. Conclusion and Perspectives

We presented a model for measuring similarity in a general Rhythm Space, which include all the possible rhythm sequences. The key issue and innovative contribution of this work is the hierarchical subdivision of rhythm sequences in several levels and the construction of a Distance Matrix for each one of them. The information is coded in a Similarity Coefficient Vector (SCV), whose entries estimate the similarity between rhythm sequences in different k-levels. These coefficients are related to the sparsity the k-levels Distance Matrices. We also provided a easier to read single value measure for similarity. In addition we presented two applications for our formal model. Clearly, it can be applied in many other areas of music analysis and composition. It can be also applied on musical learning devices, such as self evaluation systems to relate played sequences to previously defined ones.

There is plenty of room to extend this work. For example, is yet to be done the comparison between this distance with other well established similarity measurements, such as the Levenshtein and Hausdorff distances. In addition to the block distance, we could use new basic p-distances and check which of them is better to the applications the user has in mind.

A further and also important problem is to link the formal similarity in this work to the rhythmic perception of human beings.

In this paper we have only shown the potential of our methods to construct similarity measures. The problem of perception of rhythm sequences deserves a deeper study by itself. This, as well comparisons with other methods, will be done in a future work.

6. Acknowledgments

This work was supported by the Leverhulme Trust (JM), the São Paulo State Research Foundation (FAPESP) (AMJ), and CAPES (Brazil) (MG, AMJ). A.M.J is grateful to the Computer Music Research group from the University of Plymouth, for the hospitality during the making of this work.

References

Cooper G., Meyer, L. (1963). *The Rhythmic Structure of Music*. University Of Chicago Press.

Cox, A. (2001). The mimetic hypothesis and embodied musical meaning. *MusicæScientiæ*, 2:195–212.

Dawkins, R. (1989). *The The Selfish Gene*. Oxford University Press, London.

Dawkins, R. (1991). *The Blind Watchmaker*. Penguin Books, London.

Gabora, L. M. (1997). The origin and evolution of culture and creativity. *Journal of Memetics -Evolutionary Models of Information Transmission*.

Gabrielsson, A. (1973). Similarity ratings and dimension analyses of auditory rhythm patterns i. *Scand. J. Psychol.*, 14:138–160.

Handel, S. (1989). *Listening: An Introduction to the Perception of Auditory Events*. The MIT Press.

Hewlett, W. B. and Selfridge-Field, E., editors (2005). *Music Query: Methods, Models, and User Studies*. MIT Press.

James, D. L. and Miikkulainen, R. (1995). SARDNET: a self-organizing feature map for sequences. In Tesauro, G., Touretzky, D., and Leen, T., editors, *Advances in Neural Information Processing Systems 7*, pages 577–84, Cambridge, MA, USA. MIT Press.

Kohonen, T. (1985). The self-organizing map. In *Proceedings of IEEE*, volume 78, pages 1464–1480.

Minsky, M. (1988). *Society of Mind*. Simon & Schuster.

Miranda, E. R. (2004). At the crossroads of evolutionary computation and music: Self-programming synthesizers, swarm orchestras and the origins of melody. *Evolutionary Computation*, 12(2):137–158.

Mongeau, M. and Sankoff, D. (1990). Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175.

Orpen, K. S. and Huron, D. (1992). Measurement of similarity in music: A quantitative approach for non-parametric representations. *Computers in Music Research*, 4:1–44.

Povel, D. J. and Essens, P. (1985). Perception of temporal patterns. *Music Perception*, 2(4):411–441.

Takeda, H., Nishimoto, T., and Sagayama, S. (2003). Automatic rhythm transcription from multiphonic midi signals. In *ISMIR 2003*.