STRUCTURING MUSIC BY MEANS OF AUDIO CLUSTERING AND GRAPH SEARCH ALGORITHMS

Frédéric Le Bel

STRUCTURING MUSIC BY MEANS OF AUDIO CLUSTERING AND GRAPH SEARCH ALGORITHMS

Frédéric LE BEL – frdric.lebel@gmail.com
Centre de recherche en informatique et création musicale, Laboratoire MUSIDANSE – Université Paris 8,
Représentations musicales, IRCAM – Centre Pompidou,
Paris – France, 2017.

RÉSUMÉ

Cet article propose d'explorer un 'work in progress' cherchant à développer une démarche de composition assistée par ordinateur (CAO) axée sur l'élaboration de structures musicales au moyen d'une méthode de classification audio non supervisée et de certains algorithmes de parcours de graphe. Certaines parties de cette idée ont déjà été étudiées, notamment dans le cadre de la synthèse concaténative par corpus, de la reconnaissance des genres musicaux ou de l'orchestration assistée par ordinateur pour en nommer quelques-unes, mais le défi reste de trouver une manière d'intégrer ces techniques informatiques dans le processus compositionnel, non pas pour générer du matériau sonore mais plutôt pour l'explorer, pour l'analyser et pour mieux le comprendre en préparation d'une œuvre musicale. Contrairement aux démarches traditionnelles de CAO, principalement axées sur des méthodes génératives, la suivante propose une approche analytique du matériau sonore axée sur l'élaboration de différents types de structures musicales afin de stimuler le processus créatif. Cet article propose donc de disséquer la structure algorithmique afin d'exposer la méthodologie, d'éclairer certaines problématiques inhérentes, d'exposer les limites d'une telle approche et finalement de présenter quelques idées pour d'autres applications.

1. INTRODUCTION

This article proposes to explore a work in progress that aims at developing a computer-aided-composition (CAC) approach to structuring music by means of audio clustering and graph search algorithms. Although parts of this idea have been investigated in order to achieve different tasks such as corpus-based concatenative synthesis [1], musical genre recognition [2] or computer-aided orchestration [3] to name a few, the challenge remains to find a way of integrating these techniques into the creative process; not to generate material but to explore, to analyse and to understand the full potential of a given sound corpus prior to scoring a musical piece. As opposed to mainstream CAC tools, mostly focusing on generative methods, the following one proposes an analytical approach to structuring music through the creative process. As the title of this article reveals some pieces of answer to this interrogation, the following topics aim at unfolding the algorithmic structure in order to examine the methodology, to discuss a few important co-lateral problematics, to expose the limitations of such an approach and finally to discuss ideas for future development.

2. STRUCTURAL OVERVIEW

The algorithmic structure may be divided into three distinct processing stages. The first one, including the signal modelling, the audio features extraction and the temporal modelling, focuses on data extraction. The second stage, including the features space analysis, the calculation of a distance threshold and the audio clustering itself, focuses on data analysis. The third one, including the clusters space analysis and the graph exploration, focuses on data sorting.

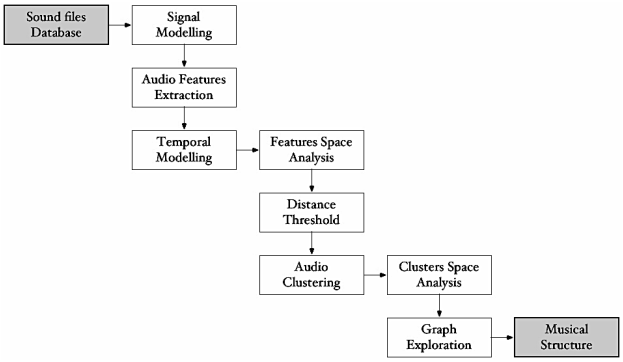

Figure 1. Structural overview

3. DATA EXTRACTION

3.1 Signal Modelling

Considering that the database is composed of pre-segmented sounds of any lengths (user defined), the process starts with a very simple but crucial procedure which is to prepare the sound files for audio features extraction by applying different types of processing. The goal is to optimize the input data in order to extract relevant information for the algorithm to detect significant patterns. Somehow, it is to mimic the human selective listening skill by reducing the information to what is perceptually consistent in the audio files.

In the frame of this work, the sound files are systematically resampled to 44.1kHz/16bits before the audio features extraction. Then, depending on the source of the recording (analogic or digital), different treatments may be applied in order to smooth and clean the signal such as filters, de-clip, de-click, hum removal, de-noise and spectral repair. Also, each sound file is mixed down to a single channel in order to remove the sound localization variable. Finally, the tracks are normalized to 0dB with-

out any type of compression in order to amplify the data without removing the relative amplitude of each frame. These processing are applied for the sake of analysis only[1]. The final output may use the original sound files.

3.2 Audio Features Extraction

The audio features extraction consists of decomposing the sounds into specific properties based on the energy, the spectrum and the harmonic content, either from a physical or a perceptual model applied to a raw signal. As one may listen to the same sound from different perspectives, segregating the different components, the idea is to project this ability into a computerized sound analysis.

In the frame of this work, different engines are used in order to extract different types of data. For low-level features, the Ircamdescriptors-2.8.6 [4] appears to be very efficient. For others such as partial tracking and chord sequence analysis, the pm2 engine [5] is used. Also, the superVP engine [6] is used to extract the peak analysis and the masking effects. Without being exhaustive, the following selection of audio features covers a wide range of the complexity of sounds for the user to make custom sub-selections. Under two different models (physical and perceptual), the low-level descriptors may be separated into four categories: instantaneous temporal, instantaneous harmonic, instantaneous energy and instantaneous spectral.

Physical model	Perceptual model
Instantaneous temporal descriptors	
• Auto-correlation • Signal zero crossing rate	
Instantaneous energy descriptors	
	• Loudness • Spread • Relative Specific Loudness
Instantaneous spectral descriptors	
	• MFCC • Spectral centroid • Spectral spread • Spectral skewness • Spectral kurtosis • Spectral decrease • Spectral roll-off • Spectral variation • Spectral deviation • Spectral flatness • Spectral crest
Instantaneous harmonic descriptors	
• Fundamental frequency • Inharmonicity • Noisiness • Chroma • Chord sequence analysis • Partial tracking • Peak analysis • Masking effects	• Harmonic spectral deviation • Odd to even energy ratio • Tristimulus

Table 1. Selected low-level descriptors

[1] Obviously, this procedure may vary a lot according to the database. The idea is to clearly determine what is perceptually consistent or what is to be analysed into the clustering process and prepare the audio files consequently.

The two models, physical and perceptual, imply a pre-processing stage in order to provide the adequate signal representations for computing the descriptors. In both cases, but depending on the feature to extract, the pre-processing may consist of:

- Energy envelope estimation (sampling),
- Temporal segmentation (windowing),
- Short-time Fourier transform [7],
- Harmonic sinusoid model [8].

The perceptual model differs from the other one because it implies an additional set of pre-processing that attempts to emulate the human auditory system. This specific chain of processing consists of a mid-ear filtering [9] and a logarithmic band conversion, namely the Mel bands [10], used for the MFCCs, and the Bark bands [11], used for all others.

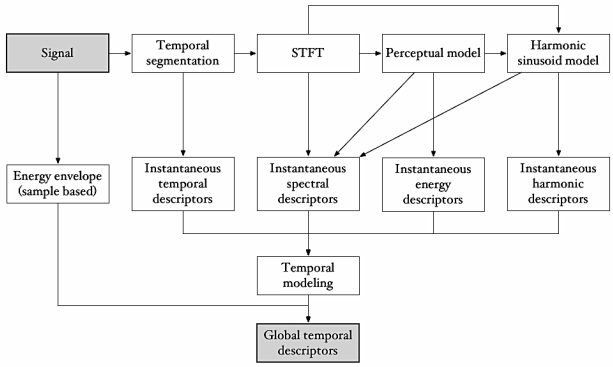

Figure 2. Audio features extraction flowchart

As shown in figure 2, the spectral descriptors may be computed directly over the STFT but also over the perceptual model and the harmonic sinusoid model. In the frame of this work, as seen in Table 1, the spectral descriptors are computed over the perceptual model only. It is also shown that the harmonic descriptors may be computed over the physical (STFT) or the perceptual model as for the harmonic spectral deviation, the odd to even energy ratio and the tristimulus. As shown in Table 1, both approaches are used in the frame of this work.

3.3 Temporal Modelling

The temporal modelling is similar to the signal modelling. It consists of reducing the information to what is perceptually consistent in the audio features (not the sound files) by applying different processing. Here again, the idea is somehow to mimic the selective listening skill. In this sense, the instantaneous features, except the ones expressed in frequencies (Hz), can be weighted by the energy (amplitude^2) of their corresponding signal (user defined) and smoothed by the use of a 5^{th} order median filter [12]. Although different windowing patterns are possible for multidimensional features (box and cross patterns), the temporal modelling is applied, if so, independently on each coefficient. Thus, each of them is considered as an individual time series and is treated consequently for more accuracy.

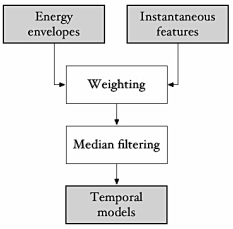

Figure 3. Temporal modelling flowchart

Then, for mathematical reasons (calculation of the Cosine similarity and the Spearman coefficient), the audio features are resized for their lengths to match when later compared pairwise.[2] Hence, for each pair of audio features that is compared, the shortest is oversampled to fit the size of the longest. From a perceptual angle that is to say, if two sounds are identical, except for their durations, they are perceived as equivalent or extremely similar. Below is an example of two superimposed audio features of the same type (y = spectral centroid) calculated over two different sounds of different durations (x = time).

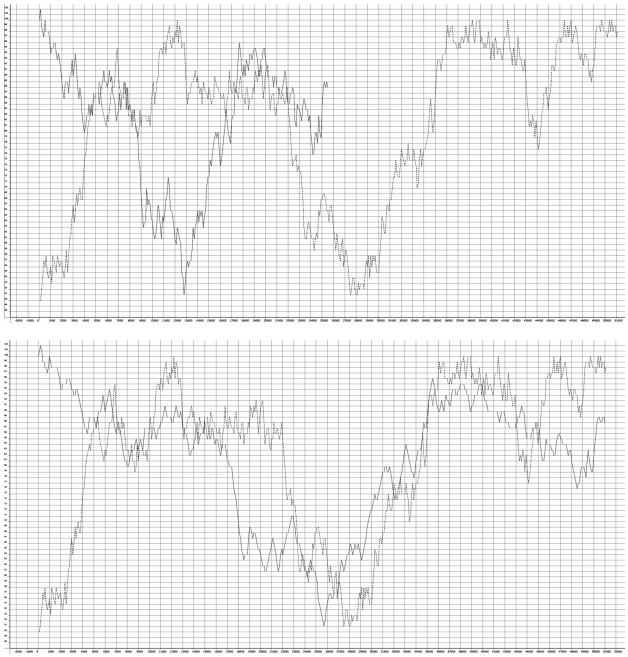

Figure 4. [top] Original features, [bottom] Resampled features

From figure 4 (top) they seem to be very different considering their original lengths but, from figure 4 (bottom), they seem a lot more alike considering that the shortest was oversampled to match the length of the longest. Although the previous approach seems to fit reality, this question should be further investigated from a perceptual angle.

[2] For this kind of temporal alignment, multiple approaches were tested. One was to pad the shortest feature from a pair with a given numerical value but then, the comparison was affected by the added values. Another one was to use the fast dynamic time warping (fastDTW) [13] but then, the comparison was biased by the overfitting yielded by the latter method. Considering the previous drawbacks, a classic resampling method then appeared to be the way with minimum impact on further calculations.

4. DATA ANALYSIS

4.1 Features Space Analysis

Following the data extraction is to determine the level of similarity, or dissimilarity, between each component of the database taken pairwise.[3] Considering that only instantaneous features (time series) are taken into account, three different approaches may be adopted, all combinations included (user defined), to process such data structures. The first approach is based on distances (magnitudes), the second one is based on similarities (orientations) and the third one is based on correlations (dependencies).[4]

4.1.1 Mahalanobis distance (magnitude)

The Mahalanobis distance is a measure of the distance between a point P and a distribution D. It is a multidimensional generalization of the idea of measuring how many standard deviations away P is from the mean D. This distance is zero if P is at the mean of D, and grows as P moves away from the mean. This type of distance is thus unit less, scale-invariant and takes into account the variance and the correlation of the data set [14].

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T\, S^{-1}\, (\vec{x}-\vec{y})} \qquad (1)$$

4.1.2 Cosine similarity (orientation)

The cosine similarity is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them. The cosine of 0° is 1, and it is less than 1 for any other angle. It is thus a judgment of orientation and not of magnitude. For example, two vectors with the same orientation have a cosine similarity of 1, two vectors at 90° have a similarity of 0, and two vectors diametrically opposed (180°) have a similarity of -1, independent of their magnitude (translation-invariant) [15].

$$cos(\theta) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}} \qquad (2)$$

4.1.3 Spearman's rank correlation (dependencies)

Spearman's rank correlation assesses monotonic relationships whether linear or not. A perfect Spearman correlation of +1 or −1 occurs when each of the variables is a perfect monotone function of the other. Intuitively, the Spearman correlation between two variables is high when observations have a similar rank between the two variables, and low when observations have a dissimilar rank between the two variables. A value of zero denotes a total independence (zero relationship) between the rank of the two variables. Unlike the cosine similarity, this type of coefficient gives a clear indication about the direction of

[3] As discussed earlier, this can be done using one or more of their audio features in order to create customized perspectives on sound materials through clustering (user defined).

[4] As demonstrated further, the previous approaches may also be merged in order to account for higher level descriptions of sounds into the clustering process.

the function. It is also translation-invariant [16].

$$\text{tie corrected } \rho = \frac{(n^3-n)-6\sum_{i=1}^{n} d_i^2-(T_R+T_S)/2}{\sqrt{(n^3-n)^2-(T_R+T_S)(n^3-n)+T_R T_S}} \quad (3)$$

where,

- n is the length of the variable,
- R and S are the rank variables,
- d is the difference between R_i and S_i,
- T_R and T_S are the tie-correction factors.

$$T_R = \sum_{g=1}^{G}\left(t_g^3 - t_g\right) \quad (4)$$

where,

- G is the distinct values of the averaged ranks,
- t_g is the number of occurrence of a distinct rank.

4.1.4 Distance triangulation

If using more than one of the three types of measurements (distance, similarity and correlation), the following approach is to merge them into a single multidimensional score. Since each type describes a different aspect of similarity (magnitude, orientation and dependency), the goal is to obtain a single value that includes all three perspectives. Based on the concept of triangulation, the principle is to project the previous measurements in a common space where each axis represents a different one of them. This allows to triangulate the resulting location for later calculating the Euclidean distance between this new coordinate (x, y, z) and the origin (0, 0, 0) [17].

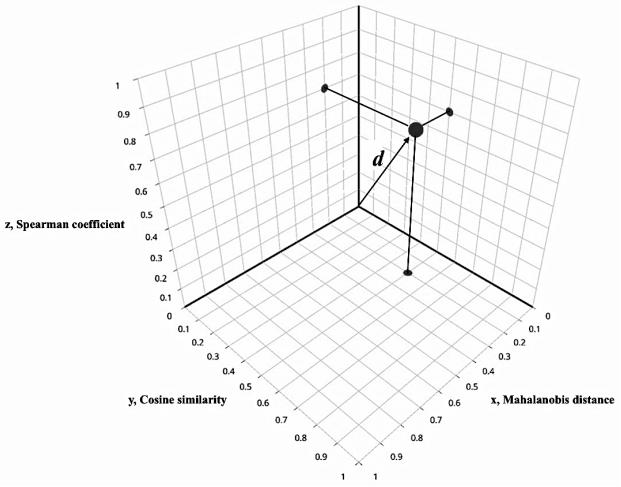

Figure 5. 3D space distance triangulation

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \quad (5)$$

For that, the measurements must be adapted to the new space for the origin to be the right target (minimum distance). Knowing that the Mahalanobis distance range may be statistically ceiled between [0. 4.], that the Cosine similarity ranges between [-1. 1.], and that the Spearman's rank correlation coefficient is bounded between [-1. 1.], it becomes obvious that they can be normalized between [0. 1.] and rescaled to [1. 0.] for fitting the previous space and for its origin to represent the minimum distance. The reason to rescale the data is to convert the degrees of similarity and correlation into distance interpretable values and the reason to normalize them is to work in an even space and avoid favouring one or another of the values while triangulating them. In a way, it is to give the same weight to all the given values.[5]

4.1.5 Distance matrices

After measuring the level of similarity between each pair of sounds upon specific audio feature(s), the resulting values are gathered inside distance matrices, each of those representing a different feature. Meaning that the level of similarity is computed between corresponding audio features only: $d(x_i, y_i)$ and not $d(x_i, y_j)$ nor $d(x_j, y_i)$. This way, each matrix includes the distance between each pair of sounds under a specific property of sounds. The resulting number of matrices is thus linked to the number of extracted features (n features = n matrices).

4.1.6 Weighted scores

For each distance matrix, thus each selected audio features, it is possible to apply different weights to them (user defined). In other words, that is to assign independent degrees of importance to the different features that are considered for the clustering. That is done by multiplying each scores within a single matrix by a weight factor (w_i) ranging from 0 to 1 [18].

4.1.7 Dimensionality reduction

If working with multiple audio features at once, the dimensionality of the resulting space (n-features = n-matrices = n-dimensions) needs to be reduced in order to process the final clustering. One of the most common method to do so is known as the principal component analysis (PCA) [19]. Although the latter technique could be used at this point, or earlier in the process for feature selection and/or feature extraction [20], it was not considered to be necessary considering that the maximum number of dimensions (n-features) used in the frame of this work is less than a hundred. Hence, there is no serious reasons to fear the curse of dimensionality [21]. Besides, it also appeared to be more consistent to keep control over the processed information for later interpreting the clustering results without adding confusion.

In this sense, the following solution is not to reduce the number of dimensions themselves but rather to project them in a common space. The idea is the same as described before (distance triangulation). Although the resulting space may be expressed in more or less than three dimensions here, the approach remains the same: calculating the Euclidean distance (5) between the new

[5] However, it should be mentioned that in the case of the Mahalanobis distance, the values do not need to be rescaled since it is already a distance measure. It should also be mentioned that in the case of the Cosine similarity and the Spearman's coefficient the absolute values are used in order to fit the Mahalanobis distance scale and avoid space fitting problems. From a perceptual angle, it means that a given sound and its inversion, or its retrograde, are considered equivalent. Similar to the temporal alignment method mentioned before, this seems to make sense but it should also be further investigated from a perceptual point of view in order to determine the level of similarity between a sound and its opposite.

coordinate (x, y, z, …, *n*) and the origin (0, 0, 0, …, 0). In other words, the multiple dimensions are triangulated and summarized by their distance to the origin (minimum distance). Consequently, this approach allows to bring down the number of distance matrices to a single one (features-space distance matrix) without applying any transformation to the original data.

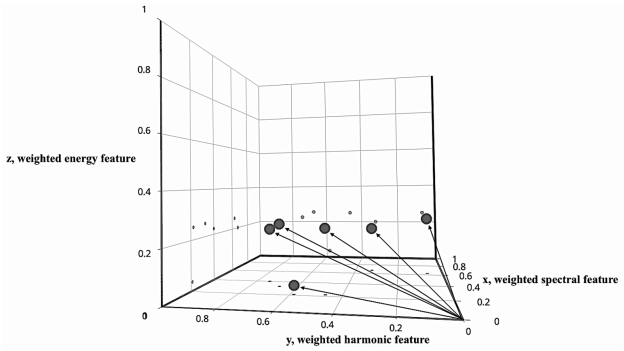

Figure 6. 3D space dimensionality reduction

In figure 6, each node represents a specific pair of sounds ((a b)(a c)(a d)(b c)(b d)(c d)) and the axes represent the distances corresponding to each feature.

4.1.8 Single-layer relational space

Although this is not an intrinsic part of the clustering process, it is interesting to mention that the database can be viewed as some kind of relational space at this point. With the help of Gephi [22], an open source software for exploring and manipulating networks, any features-space distance matrix can be visualized as such.

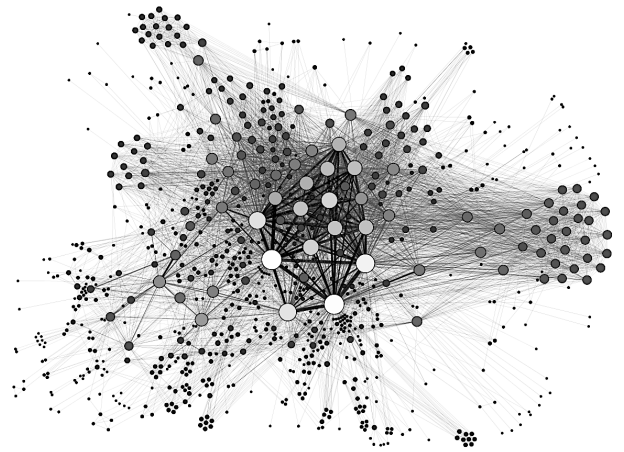

Figure 7. Features-space network

In figure 7, the nodes represent the sounds and the edges represent the distances between them. From this perspective, it is clear that the matter of this work is not about positioning the sounds in space but rather to define the strength of their respective networks.

4.2 Audio Clustering (fuzzy clustering)

The algorithm used in the frame of this work may be seen as a variation on the hierarchical cluster analysis (HCA) based on a distance threshold constraint instead of linkage criteria. Hence, instead of targeting the nearest neighbours (open space), a user defined threshold (closed space) determines the targets. The threshold represents the maximum distance between two elements to be agglomerated. It can be seen as some kind of perceptual threshold. In this sense, the targets should be as many as they fall below the maximum distance allowed. For that to be assessed, the number of iterations has to be as much as the square number of elements. In other words, each component has to be tested before defining a cluster. That means the algorithm is looking for the global optimum, including the possibility of one component belonging to multiple clusters (overlapping clusters). In this case, contrary to the greedy algorithms, the speed is traded for completeness and accuracy of the clustering. The following figure, where the radiuses of each circle represent the distance threshold as well as each of them an iteration of the process, partially demonstrates the agglomeration strategy described above.

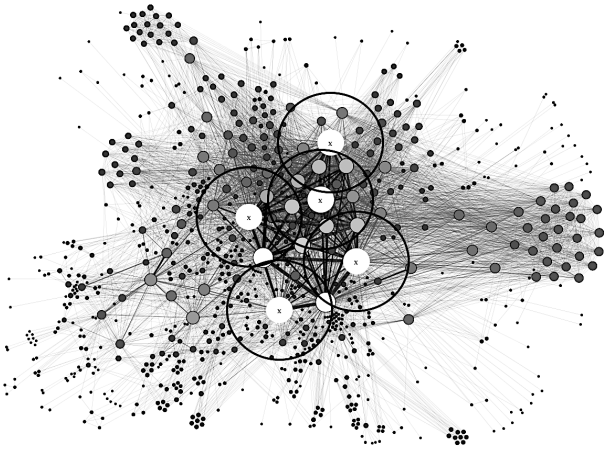

Figure 8. Fuzzy clustering in the features-space

Concretely, the distance threshold is applied to each column, or each row, of the features-space distance matrix, meaning that the centre of each cluster corresponds to a sound itself. From another angle, it also means that the space is defined by its components and not the opposite.

4.2.1 Distance threshold

Considering that the components have to fall below a user defined threshold to be agglomerated, the definition of space and the number of dimensions have a significant impact on the results, thus on setting the threshold itself [23]. In other words, the threshold is a space dependent variable (relative value) and thus should be adapted to every case.[6] Consequently, the distance threshold should be informed by some kind of sparsity analysis computed upon the features-space distance matrix. In the frame of this work, the definition of this parameter relies on a histogram based on the Freedman-Diaconis rule [24],

[6] Although the ultimate objective is to identify clusters of sounds sharing strong similarities upon particular audio features, where the distance threshold would act as some kind of perceptual witness, the reality is that datasets are inflexible. In this sense, the algorithm cannot find what do not exist.

itself based on the most significant basic robust measure of scale, the interquartile range (IQR) [25].

$$\text{bin width } h = 2\frac{\text{IQR}(x)}{\sqrt[3]{n}} \quad (6)$$

The number of bins can then be calculated from the following formula in order to build a consistent histogram upon which the user can assess the sparsity of its database and later determine a proper distance threshold [26].

$$\text{number of bins } k = \left\lceil \frac{\max x - \min x}{h} \right\rceil \quad (7)$$

4.2.2 The outcome

Using the clustering method described before leads to a particular outcome (fuzzy clustering). Being more accurate and complete that in the case of a classic HCA, the resulting space is more complex. Related to the possibility of clusters to overlap, a new genre of hierarchy appears among the elements. In a way, the hierarchy is blurred.[7] Thus, in order to simplify the data structure and to avoid duplicates with different labels, clusters within clusters (sub-clusters) are simply merged together.

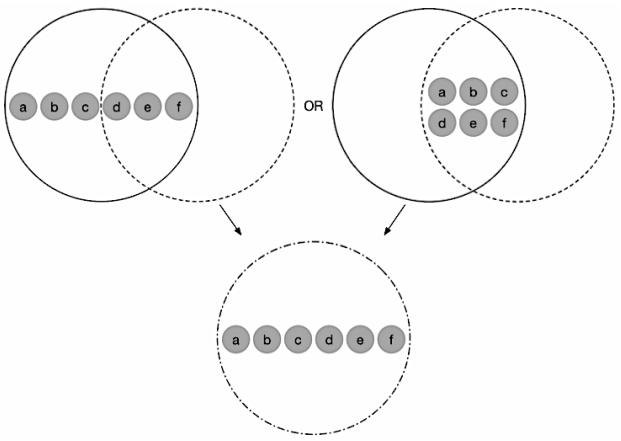

Figure 9. Sub-clusters merge

In the case where two clusters have distinguished and shared elements (overlapping clusters), they are simply considered as two different entities with common elements.

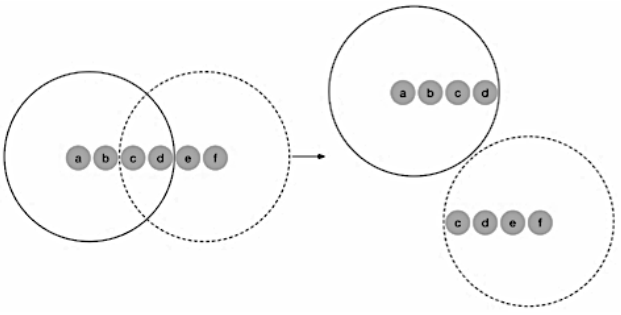

Figure 10. Overlapping clusters split

[7] That is because the overlaps allow a single element to belong to multiple clusters at once, thus making the clusters more difficult to distinguish.

Being inherent to the agglomeration strategy described earlier, this particularity becomes very interesting when translated into musical terms. In this sense, the overlaps may be seen as common notes between different chords. Such case often leads to what is known as voice-leading patterns. This technique is widely used in music to sequence various harmonic structures such as the typical (VI-II-V-I) chord progression in tonal music. This work being audio feature oriented, the previous analogy could be translated to some kind of timbre voice-leading where the overlaps, or the intersections between clusters, become pivots for commuting between two distinct groups of sounds [27].

5. DATA SORTING

5.1 Clusters Space Analysis

The clusters space analysis (post-clustering) is about gaining more insight on the resulting network. The same way as for the single-layer relational space presented earlier, the idea is to visualize the clustered space network in order to have a better understanding of its core structure and to provide crucial information for later graph exploration.

5.1.1 Intra-cluster modelling

The intra-cluster modelling consists of generalizing the features of each cluster. The idea of creating these global models is to obtain a general, but still accurate, description of every single cluster. In a way, it is to find the theoretical centre of mass or the barycentre of each cluster.[8] Since the clusters are composed of sounds, the global models are obtained by merging the corresponding audio features and by accumulating the results in a list (vector), the latter being the global model (barycentre) itself. In this case, each data point of the resulting vector is calculated as the arithmetic mean of the arithmetic means of each audio feature respectively.

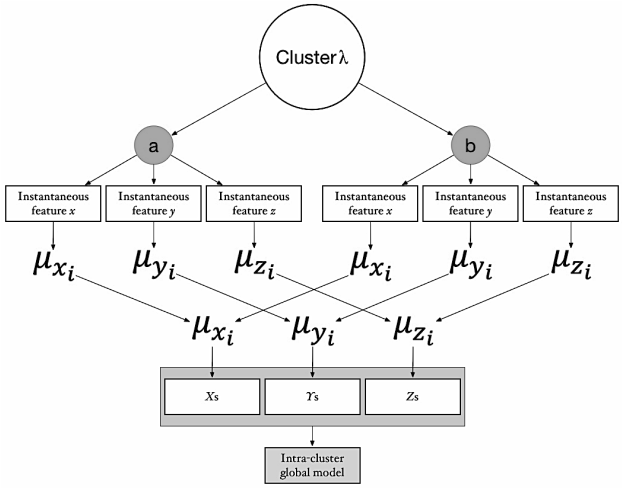

Figure 11. Intra-cluster modelling

[8] As for the previous single-layer relational space, this information is essential to outline the resulting network.

Similar to the temporal modelling, each coefficient of a multidimensional feature is considered as an individual time series and is treated consequently. Thus, each coefficient represents a different data point calculated as the mean of the means in the resulting vectors.

5.1.2 Audio clustering (Neat clustering)

At this stage, the user may decide to reformulate the clustering in a way to remove the fuzzy components, the sounds that are assigned to multiple clusters, in order to obtain a neat clustering where each sound belongs to a single cluster. That is done by measuring the distance (magnitude) between the sounds and the centre of their assigned clusters in order to identify the closest and make it the final assignation.[9] For that, each sound has to be modelled similarly to the previous intra-cluster modelling method in order to assess the distance to its clusters barycentre. Since a sound is a single element, the difference is that there is no need to average the means of each audio feature.

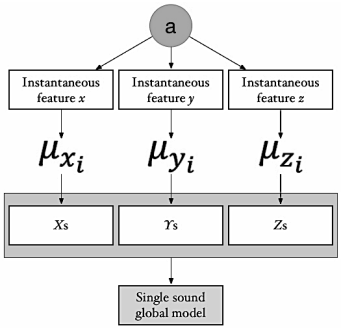

Figure 12. Single-sound modelling

Considering that both global models, the intra-cluster and the single-sound, are expressed as vectors of information, the Mahalanobis distance (1) appears to be the most appropriate measurement in this case also. Using this last clustering method, the circular shape of clusters, as seen in figure 8, may now be seen as if they were closed under vacuum from their centre.

Figure 13. Neat clustering in the features-space

[9] From a probabilistic point of view, it is to say that the closer a sound is to the centre of a cluster, the more probable it is to belong to it.

5.1.3 Inter-cluster analysis

Similar to the features-space analysis described earlier, the inter-cluster analysis consists of measuring the distance (magnitude) between each cluster's barycentre. For that, the Mahalanobis distance (1) also appears to be the most adequate measurement in order to process this type of data structure. Then, as for the single-layer relational space, a distance matrix is built from the previous calculations in order to outline the underlying network.[10]

Although this could be enough to unravel the network, it appeared to be important to add a second element in the process with respect to the fuzzy clustering approach. That is to use the Jaccard distance [28]. While the Mahalanobis distance informs a metric distance between two theoretical barycentre, the latter informs a similarity level (distance interpretable) based on the ratio of two given cluster's intersection (shared components) and their union (combined components).[11] In this case, the Mahalanobis distance [0. +inf.] is simply weighted (multiplied) by the Jaccard distance [0. 1.].

$$J\,d(A,B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{8}$$

While the latter is useful to arbitrate between two clusters having the same Mahalanobis distance to a third one, the former is useful to keep track of two others not sharing any components (no intersection) with another one.

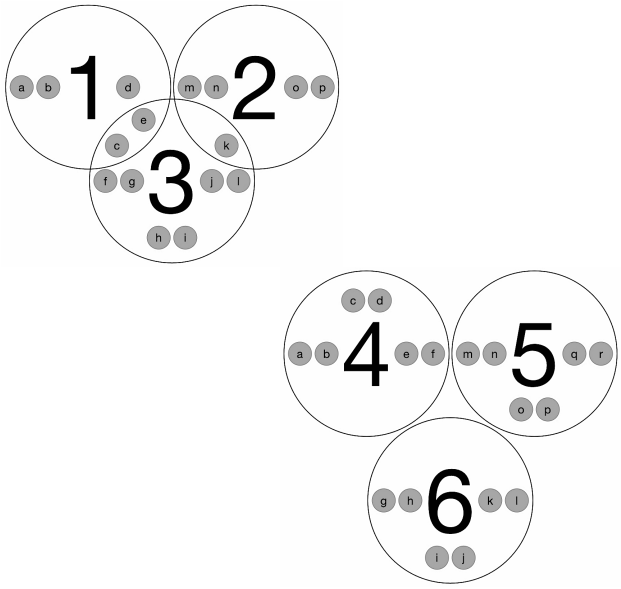

Figure 14. [top-left] Clusters with shared components, [bottom-right] Clusters without shared components

In figure 14 (top-left), if the Mahalanobis distance between cluster no.1 and cluster no.3 equals 1, and that between cluster no.2 and cluster no.3 also equals 1, but that the Jaccard distance between cluster no.1 and cluster

[10] Since it is based upon clusters of sounds (audio features), the resulting matrix may be seen as a clusters-space distance matrix by analogy to the previous features-space distance matrix.

[11] Similar to the idea of distance triangulation, here also, it is about combining two different perspectives on a unique case in order to strengthen the results of the analysis.

no.3 equals 0.82, and that between cluster no.2 and cluster no.3 equals 0.91, the conclusion is that cluster no.1 and cluster no.3 are closer (smallest distance). In the other case (figure 14 bottom-right), since there is no intersection between any of the clusters, the Jaccard distance equals 1 (maximum distance) for every pair. Thus, the Mahalanobis distance remains the only, and sufficient, measurement index to compare them.

5.1.4 Multi-layer relational space

Following the intra-cluster modelling and the inter-cluster analysis, the resulting clusters-space distance matrix can be used, also with the help of Gephi, to visualize the underlying network. In this case, the nodes represent the clusters and the edges represent the distances between them. From this perspective, the resulting relational space is multi-layered. In other words, the network is itself composed of networks. Meaning that each node (cluster) embeds a smaller network of the same type.

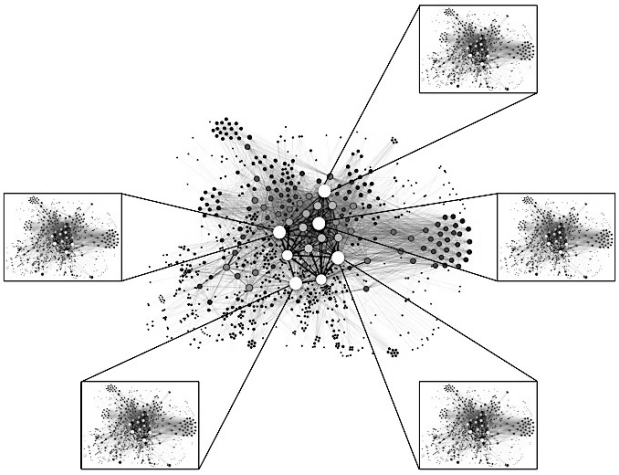

Figure 15. Clusters-space network

Regarding the overlapping clusters, it is now clear that the more they overlap, the closer they are. Consequently, this approach already suggests ways for navigating through the network (distance based) towards the previous idea of timbre voice-leading.

5.2 Graph Exploration

Far from being exhaustive, the following section proposes two simple approaches for the user to start structuring music from the previous clusters-space analysis. In this sense, the graph exploration consists of finding path(s) that could suggest way(s) of sequencing the different clusters of sounds in some kind of chronological order. Considering that the resulting network, as yielded from the clusters-space analysis, translates to a complete, undirected and weighted graph, two types of approaches seem to apply quiet naturally. The first one is to find Hamiltonian paths and the second one is to find Spanning trees.

5.2.1 Hamiltonian paths

A Hamiltonian path is verified when a trail in a graph can visit each node (cluster) exactly once. The starting and ending node may not be the same. If they are, the path is a Hamiltonian cycle.

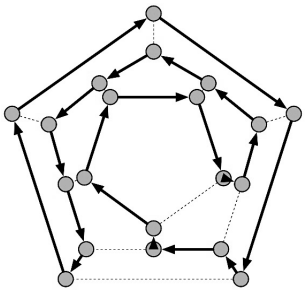

Figure 16. Hamiltonian cycle

Considering that the clusters-space graph is complete, thus having n! Hamiltonian paths, the following problem is not how to find them but rather what kind to find. In this sense, the well-known traveling salesman problem (TSP) seems to provide an interesting premise.[12] Then, from a compositional angle, the shortest Hamiltonian path would represent an ordered sequence of clusters for which the total distance is minimal. In other words, the clusters would be sorted in a way that the global similarity is maximized.

Usually, two approaches may apply to solve a TSP. The first is to use exact algorithms, which works reasonably fast for small problem sizes. The second one is to use heuristic algorithms, which deliver either seemingly or probably good solutions. Considering that the running time for exact algorithm lies within the factorial of the number of cities, this approach becomes impractical even for only 20 cities [30]. For that reason, the heuristic approach is often preferred. In the frame of this work, two types of heuristics, the nearest neighbour algorithm [31] and the Concorde TSP Solver [32], are available for the user to estimate the shortest Hamiltonian path from any given clusters-space network as described before.

5.2.2 Spanning trees

A spanning tree of an undirected graph is a subgraph that includes all of the nodes. In general, a graph may have several spanning trees. They can be defined as a maximal set of edges from the initial graph that do not contain any cycle [33], or as a minimal set of edges that connects all nodes together [34].

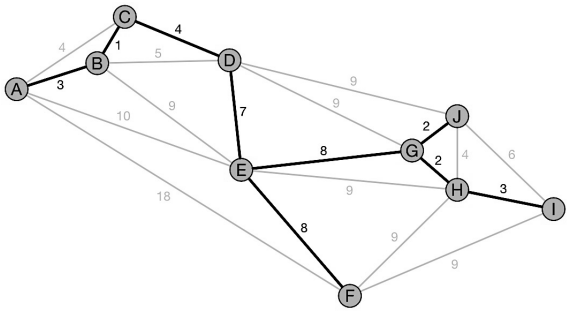

Figure 17. Spanning tree

[12] The TSP asks the following question: Given a list of cities (nodes) and the distances between each pair of them, what is the shortest possible path that visits each city exactly once and returns (or not) to the origin city? [29]

Considering that the clusters-space graph is complete, thus having n^{n-2} number of spanning trees [35], the problem is similar to as formulated before. It is not how to find them but rather what kind to find. In this sense, the minimum spanning tree (MST) seems to provide another interesting premise. Similar to the shortest Hamiltonian path, a MST is a subset of the edges of an undirected and weighted graph that connects all the nodes together with the minimum possible total edge weight without any cycles. In other words, it is a spanning tree (not a path) whose sum of edge weight is as small as possible [36]. Hence, the MST does not represent an ordered sequence of clusters but rather some sort of an optimized road map through the clusters-space from which the user can trace its own way. However, the clusters remain connected in a way that the global similarity is maximized. In the frame of this work, the Kruskal algorithm [37] is available for the user to find the minimum spanning tree for any given clusters-space network as described before.

5.2.3 Discussion

As shown in the previous section, the MST is similar to the TSP in the way that both approaches look for a subset of edges that connects all the nodes together with the minimum possible total weight (or distance). The main difference between them is that the TSP leads to a sequenced solution (path or cycle) while the MST leads to a solution that is not sequenced (no path nor cycle). Hence, they provide a different representation, thus a different point of view on the same problem. While the algorithms used to solve the TSP lead to a closed pattern solution for exploring the space, the MST leads to an open one. In a way, the latter may offer more flexibility to explore the space and to structure music. However, both approaches appear to be an interesting way for exploring the clusters-space towards the idea of timbre voice-leading discussed earlier.

Although the previous section covered only a few types of a large number of graph search methods, it is clear that there is a rich potential into using these techniques for structuring music in the continuity of audio clustering. Therefore, further investigations will be conducted in this field in order to bridge the analytical approach (audio clustering) to the compositional approach (structuring music). In this sense, it is important to mention that, in this specific context, the graph search algorithms should be used, and or developed, to solve creative problems rather than optimization ones. Actually, the whole process should be engaged with artistic purposes in order to exploit the full potential of this approach.

6. EXEMPLIFICATION

http://repmus.ircam.fr/lebel/structuring-music-by-means-of-audio-clustering-and-graph-search-algorithms

7. CONCLUSIONS

Based on previous works achieved in the field of music information retrieval such as corpus-based concatenative synthesis, musical genre recognition and computer-aided orchestration, this article exposed a different framework for audio clustering with applications to computer-aided composition. Contrary to its predecessors, this framework is built towards structuring music rather than generating sound material. In other words, it is engineered to act on a larger scale than in the other cases. Consequently, the method is designed in an attempt to render this level of perspective through the different processing stages.

7.1 Latent Problematics

As discussed along this article, many questions related to sound perception remain open despite solutions are put forward. Among those, the approach to temporal alignment (section 3.3) should be further investigated in order to have a better understanding on the effect of time (sound durations) through sound perception for measuring the similarity between sounds with more accuracy. Another one is the method for measuring the similarity itself. Using more than one approach simultaneously (magnitude, orientation and dependency), as in the frame of this work, seems to be a fairly good solution but the problem of interpreting the results accurately remains open. More specifically when comparing a sound and its inversion, or its retrograde as discussed earlier (section 4.1.4). Another problem is the shape of space implied by the distance triangulation (section 4.1.4) and the dimensionality reduction (section 4.1.7) methods discussed earlier and their impact on the shape of clusters. Although the Euclidean space seems to be well suited for achieving such tasks in the frame of this work, this question is another one that should be further investigated from a perceptual angle. Another one is related to the graph exploration. Considering that the context of this work is art oriented, the graph search algorithms should be further investigated from a perceptual angle rather than an optimization one in order to exploit the full potential of these tools into the creative process. In this sense, these algorithms should be further evaluated for their musical affect rather than for their efficiency. In other words, the question is about the kind of musical structure the various graph search algorithms may lead to.

7.2 Notable Limitations

Obviously, this approach comes with a certain lot of limitations regarding the type of input, the consequent space of variables, the clustering method itself and its manipulation. The first notable limitation is thus about using raw audio signal as main input. Contrary to mainstream CAC approaches, it may be seen as an advantage but it is actually its main disadvantage because the quality of the output is inevitably correlated to the quality of the input and also because the whole process depends on it. As discussed earlier, the signal modelling may then require a lot of time (section 3.1). Another limitation is related to the use of low-level audio features. Although the resulting space of variables may quickly become very complex and give the impression of covering a very large spectrum of sounds, the results remain interpretable on a low-level basis only, meaning that no aesthetical nor emotional affects may be considered using such an approach. Then, the clustering method is itself another no-

table limitation. Based on unsupervised learning, the method does not provide any information on the clusters components other than a similarity index. In other words, the results are simply quantitative and not qualitative. Hence, the interpretation remains confined to low-level concepts. Finally, the complexity of this tool may be a limitation itself. For instance, an 'uneducated' user may end spending a lot of time understanding the multiple parameters of this approach and their impacts on audio clustering. However, repeated experiments may lead to develop a different way of listening to sounds and eventually to use this tool as an ear trainer instead as for straight forward audio clustering.

7.3 Future Works

As mentioned before, the most important addition to this framework concerns the graph search strategies and related algorithms. Consequently, this topic is in the priority queue for future works. Also, different audio features are planned to be added and/or developed in order to offer a wider range of variables for deeper audio clustering. Besides improving the current framework, other applications and approaches are also on the way for development. Another application of audio clustering that is being investigated at the moment, using the same framework, is for musical structure analysis/extraction directly from audio signals. Also, a similar framework, based on semi-supervised learning, is under development for the user to create its own invariant models from which the clustering can be computed. In a way, the latter is thought to overcome the limitation of low-level interpretation by integrating higher-level variables. In this case, the objective is not to outline a quantitative network but rather to create different qualitative scales in order to provide the user with a higher level understanding of its sounds.

7.4 Backend discussion

The advent of electricity, in its relation to sound, has permanently modified the way of perceiving sounds and consequently deeply expanded the notion of musical meaning. The sensibility of the listener, like the composer or the performer, has then changed. Since decades, the ear has been sensitized to the subtleties of sounds and has been led to discover meaning in the qualities of new sonic territories. The continuous nature of these qualities encourages henceforth to reason on the motion of sound and to consider sound events as a state of it. In this sense, it may be fair to say that there are no more isolated objects such as melodic cells, rhythmic patterns, orchestral timbres or dynamic reinforcements, but only different states in a perpetual sound flux. Then, the concept of pattern, as it has been used in much of the music from the last centuries becomes insufficient to explain this conception of sound. What may be of interest to the contemporary composer is therefore not necessarily to create new musical objects upon which musical thoughts or actions may be directed, but may be to understand how to use those that already exist and what they can be used for. Is this postmodernity or postmodernism? No, for it proposes neither a disillusioned attitude of a satisfied epicureanism as the former, nor revisiting musical objects from the past as the latter. It is rather a question of inventing new uses for these objects, and of establishing original relationships between them. What must be the basis of musical reflection is perhaps no longer the object or the material, but the relationship. Temporal, energetic, harmonic and spectral relationships are examples. Hence, is it possible to elaborate musical works based, not on patterns proliferation or similar techniques, but rather on relationships that bring together different states of the sound continuum? Perhaps this is the moment to try to understand the world as it is instead of forcing it into a fantasised order…

8. REFERENCES

[1] D. Schwarz, G. Beller et al., "Real-time Corpus-based Concatenative Synthesis with Catart", Proc. Of the 9th Int. Conference on Digital Audio Effects, Canada, 2006.

[2] G. Peeters, "A Generic System for Audio Indexing: Application to Speech/Music Segmentation and Music Genre Recognition", Proc. Of the 10th Int. Conference on Digital Audio Effects, France, 2007.

[3] G. Carpentier, "Approche computationelle de l'orchestration musicale: Optimisation multicritère sous contraintes de combinaisons instrumentales dans de grandes banques de sons", Thèse de doctorat, Université Paris VI – Pierre et Marie Curie, Paris, France, 2008.

[4] anasynth.ircam.fr/home/english/software/ircamdescriptor-c-lib-exe

[5] anasynth.ircam.fr/home/english/software/pm2

[6] anasynth.ircam.fr/home/english/software/supervp

[7] J. W. Cooley and J. W. Tuckey, "An Algorithm for the Machine Calculation of Complex Fourier Series", Mathematics of Computation, Vol. 19, No. 90, pp. 297-301, 1965.

[8] P. Depalle, G. Garcia, et al., "Tracking of Partials for Additive Sound Synthesis using Hidden Markov Models", Proceedings of the IEEE-ICASSP, Mineapolis, USA, 1993.

[9] B. C. J. Moore, B. R. Glasberg and T. Baer, "A Model for the Prediction of Threshold, Loudness, and Partial Loudness", Departement of Experimental Psychology, University of Cambridge, UK, pp. 224-240, 1997.

[10] L. Rabiner and B-H. Juang, "Fundamentals of Speech Recognition", Prentice-Hall, New York, USA, 1993.

[11] E. Zwicker and H. Fastl, “Psychoacoustics: Fatcs and Models”, Springer-Verlag Berlin Heidelberg, 2007.

[12] T. Huang, G. Yang and G. Tang, “A fast two-dimensional median filtering algorithm”, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 27, no. 1, pp. 13-18, 1979.

[13] S. Slavador and P. Chan, “FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space”, Department of Computer Sciences, Florida Institute of Technology, USA, 2012.

[14] P. C. Mahalanobis, “On the generalized distance in statistics”, Proceedings of the National Institute of Sciences of India, Vol. 2, No. 1, pp. 49-55, 1936.

[15] A. Singhal, “Modern Information Retrieval: A Brief Overview”, Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, Vol. 24, pp. 35-43, 2001.

[16] R. Rakotomalala, “Analyse de correlation: Étude des dépendences – Variables quantitatives”, Version 1.1, Université Lumière Lyon 2, Lyon, France, 2015.

[17] J-L. Verley, “Dictionnaire des Mathématiques. Algèbre, analyse, géométrie”, Encyclopedia universalis, Albin Michel, Paris, France, 1997.

[18] J. E. Gentle, “Matrix Algebra: Theory, Computations, and Applications in Statistics”, Springer-Verlag New York, 2007.

[19] S. T. Roweis and L. K. Saul, “Nonlinear Dimensionality Reduction by Locally Linear Embedding”, Science, Vol. 290, Issue 5500, pp. 2323-2326, 2000.

[20] P. Pudil and J. Novovocova, “Novel Methods for Feature Subset Selection with Respect to Problem Knowledge”, Feature Extraction, Construction and Selection: A Data Mining Perspective, The Springer International Series in Engineering and Computer Science, Vol. 453, Springer US, pp. 101-116, 1998.

[21] R. E. Bellman, “Dynamic programming”, Princeton University Press, Princeton, New Jersey, USA, 1957.

[22] M. Bastian, S. Heymann, M. Jacomy, “Gephi: an open source software for exploring and manipulating networks”, International AAAI Conference on Weblogs and Social Media, 2009.

[23] F. Le Bel, “Agglomertive Clustering for Audio Classification Using Low-level Descriptors”, Reasearch report, version 1.0, Ircam, Paris, France, 2016.

[24] D. Freedman and O. Diaconis, “On the histogram as a density estimator”, Zeitschrigt für Wahrscheinlichkeitstheorie und verwandte Gebiete, Vol. 57, Issue 453, 1981.

[25] P. J. Rousseeuw and C. Croux, “Alternatives to the Median Absolute Deviation”, Journal of the American Statistical Association, Vol. 88, pp.1273-1283, 1993.

[26] W. N. Venables and B. D. Ripley, “Modern Applied Statistics with S”, Statistics and Computing, Springer-Verlag New York, 2002.

[27] D. Huron, “Tone and Voice: A Derivation of the Rules of Voice-leading from Perceptual Principles”, Music Perception, Vol. 19, No. 1, pp. 1-64, 2001.

[28] P. Jaccard, “Distribution de la flore alpine dans le bassin des Dranses et dans quelques regions voisines”, Bulletin de la Société Vaudoise des Sciences Naturelles, Vol. 37, pp. 241-272, 1901.

[29] K. Menger, “Das botenproblem”, in Ergebnisse eines Mathematischen Kolloquiums, Vol. 2 (K. Menger, editor), Teubner, Leipzig, pp. 11-12, 1932.

[30] R. Bellman, “Dynamic Programming Treatment of the Travelling Salesman Problem”, Journal of the Association for Computing Machinery, Vol. 9, Issue 1, New-York, USA, pp. 61-63, 1962.

[31] G. Gutin, A. Yeo, A. Zverovich, “Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP”, Disrcrete Applied Mathematics, Vol. 117, Issues 1-3, pp. 81-86, 2002.

[32] D. L. Applegate, R. M. Bixby, V. Chvatal, W. J. Cook, “The Travelling Salesman Problem”, Princeton Series in Applied Mathematics, Princeton University Press, New Jersey, USA, 2006.

[33] B. Bollobas, “Modern Graph Theory”, Graduate Texts in Mathematics, Vol. 184, Springer-Verlag New York, p. 350, 1998.

[34] P. J. Cameron, “Combinatorics: Topics, Techniques, Algorithms”, Cambridge University Press, Cambridge, UK, p. 163, 1994.

[35] A. Cayley, “A Theorem on Trees”, The Quarterly Journal of Mathematics, Oxford University Press, Vol. 23, pp. 376-378, 1889.

[36] R. L. Graham and P. Hell, “On the History of the Minimum Spanning Tree Problem”, Annals of the History of Computing, Vol. 7, No. 1, 1985.

[37] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem”, Proceedings of the American Mathematical Society, Vol. 7, pp. 48-50, 1956.