Unsupervised Music Genre Classification with a Model-Based Approach

Luís Barreira, Sofia Cavaco, and Joaquim Ferreira da Silva

CITI, Departamento de Informática
Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa
2829-516 Caparica, Portugal
lfbarreira@gmail.com, {sc,jfs}@di.fct.unl.pt

Abstract. New music genres emerge constantly resulting from the influence of existing genres and other factors. In this paper we propose a data-driven approach which is able to cluster and classify music samples according to their type/category. The clustering method uses no previous knowledge on the genre of the individual samples or on the number of genres present in the dataset. This way, music *tagging* is not imposed by the users' subjective knowledge about music genres, which may also be outdated. This method follows a model-based approach to group music samples into different clusters only based on their audio features, achieving a perfect clustering accuracy (100%) when tested with 4 music genres. Once the clusters are learned, the classification method can categorize new music samples according to the previously learned created groups. By using Mahalanobis distance, this method is not restricted to spherical clusters, achieving promising classification rates: 82%.

Keywords: automatic music genre classification, audio indexing, unsupervised classification.

1 Introduction

Since today's digital content development triggered the massive use of digital music, an indexing process is very important to guarantee a correct organization of huge databases. While a music genre categorization would be a solution, this may be hard to achieve (manually): on the one hand, music can be associated to one or more musical genres, and on the other hand, cultural differences and human interpretations, make it difficult the attainment of common music genre taxonomy. For example, an expert could label the Gustav Mahler's 2nd symphony as *Erudite - late Romantic*, while a non-expert could label it as *Classical*. Alternatively, an automatic classification based on good audio features may prevent the occurrence of incoherencies related to manual labeling.

While many supervised automatic music genre classifiers have been proposed, these will always be dependent on a previous manual labeling of the

data [4; 5; 6; 8; 9; 12; 14]. As a consequence, these will be unable to evolve with the data and automatically build new clusters driven by new values in the features. Alternatively, an unsupervised approach would not have this dependency and would be able to determine the genre of the music samples only based on their audio features. Nonetheless, only a few unsupervised methods have been proposed. Rauber et al. [10] proposed a growing hierarchical self-organizing map (which is a popular unsupervised artificial neural network) with psycho-acoustic features (loudness and rhythm) to obtain a hierarchical structuring music tree. Shao et al. [11] proposed an unsupervised clustering method that fed rhythmic content, Mel-frequency cepstral coefficients (MFCCs), linear prediction coefficients and delta and acceleration values (improvements in feature extraction) to a hidden Markov model.

Here we propose not only a methodology for unsupervised clustering but also for automatic music genre classification. The clustering method consists of a learning process that is able to cluster music samples based only on their audio properties and uses no previous knowledge on the genre of the training music samples. In addition a Model-Based approach is followed to generate clusters as we do not provide any information about the number of genres in the data set. The features used are related with rhythm analysis, timbre, and melody, among others. As these features represent a large number of dimensions, a feature reduction technique is necessary to reduce the dimensionality of the data. This clustering method achieves 100% accuracy results with classical, fado, metal and reggae music samples. After the clustering process is complete, the classification method can associate new test music samples to the previously created clusters. For that, the classifier uses Mahalanobis distance so that it can consider clusters with different shapes, volumes and orientations. An accuracy of 82% is achieved when classifying new music samples.

In the next section (Feature Extraction) we describe the features we use. Section 3 explains the Clustering Method while section 4 explains the Classification Method. The Results and Conclusions and Future Work are discussed in sections 5 and 6.

2 Feature Extraction

Feature extraction is the first step to be achieved in both automatic music genre clustering and classification. In this section, we describe the features we used, which can be grouped into two distinct groups: computational features (which do not represent any musical meaning and only describe a mathematical analysis over a signal) and perceptual features (which mathematically represent music properties based on the human hearing system).

Since some of the features we used have a very high dimensionality and it is more efficient to describe them with less dimensions, we used a set of statistical spectrum descriptors (SSD) proposed by Lidy and Rauber [7]. This set of descriptors includes: the mean, median, variance, skewness, kurtosis, min and

max-values. (Whenever this property is calculated, we mention it in the text below.)

Computational features are very popular and have been used in many automatic music genre classification studies [3; 4; 5; 6; 8; 9; 14]. To start with we use a set of *timbral texture features* proposed by Tzanetakis and Cook [14]. These include: spectral centroid (which is a measure of the centre of gravity of the magnitude spectrum), spectral roll-off (which corresponds to the frequency below which there is 85% of the energy of the magnitude spectrum), spectral flux (which accounts for the energy difference between successive frames of the spectrogram), zero-crossing rate (ZCR) (which is a measure of the number of times the audio waveform crosses the x-axis per time unit), and low energy (which is the percentage of frames that have lower energy than the average energy over the whole signal).

We also use the SSD of the MFCCs. The MFCCs are a very popular set of features based on the auditive human system that uses a Mel-frequency scale to group the frequency bins. In addition, three other features were also calculated: the root mean square of the spectrograms, which is an approximation of the volume (i.e., loudness) of the signal, the bandwidth, an energy-weighted standard deviation which measures the frequency range of the signal, and the uniformity, which measures the similarity of the energy levels in the frequency bands [3].

The spectral properties mentioned above can follow two different approaches: their values can be calculated over each window of a spectrogram or they can be calculated directly over the spectrum of the whole sound. Usually, these values are calculated over each window of the spectrogram, and that is the approach we used here. In addition, whenever we obtain a set of values with significant dimension (and we do not use their SSDs), we also use means and variances as features.

The perceptual features we used include rhythmic content, rhythm patterns and pitch content. The rhythmic content contains information such as the beat, the tempo, the regularity of the rhythm and time signature. In particular, the beat has been used in several studies on genre classification [4; 6] and it can be extracted from the beat histogram [14]. On the other hand, rhythm patterns represent the loudness sensation for several frequency bands in a time-invariant frequency representation [7]. We use both the SSDs and the rhythm histogram of the rhythm patterns. Finally, the pitch content is used to describe melody and harmony of a music signal. This feature is used quite often in genre classification leading to good accuracy results [4; 6; 13; 14; 15]. The pitch content can be extracted from the pitch histogram [15], and it includes the amplitudes and periods of the highest peaks in the histogram, pitch intervals between the two most prominent peaks and the overall sums of the histograms.

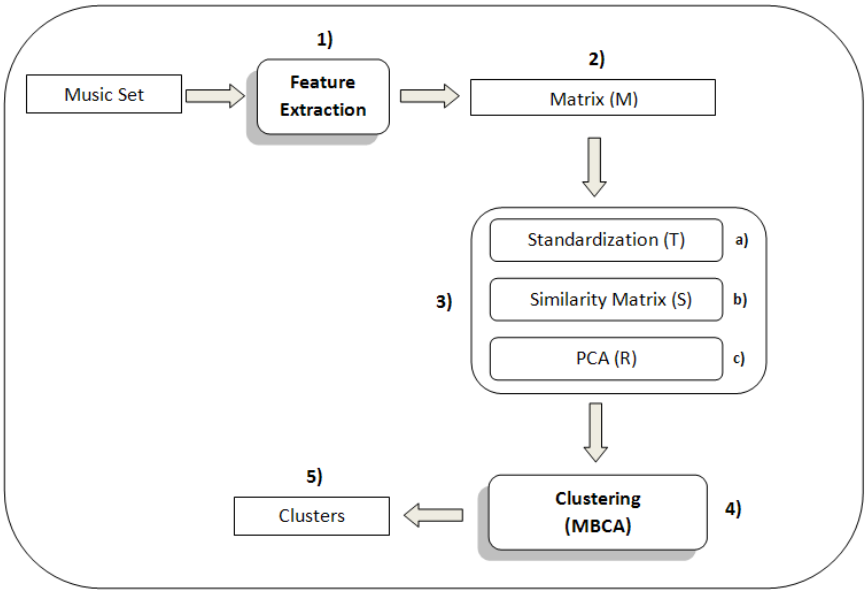


Fig. 1. The clustering method

3 The Clustering Method

The clustering method aims to organize several music samples into clusters without any initial information besides the feature set values of these samples. This method consists of several steps as illustrated in Fig. 1, which we describe below.

3.1 The Feature Reduction Stage

After the audio features have been extracted, they have to be analysed to find clusters of points with similar values. For that, the method starts by representing the features by data matrix $\boldsymbol{M}$, whose lines correspond to music samples in the training set and whose columns correspond to features. So, the $m_{s,f}$ cell of $\boldsymbol{M}$ contains the value of the fth feature for music sample s.

Once this matrix is built, the method performs some transformations as illustrated by box 3 in Fig. 1. In order to set equal importance (scale) to all columns (features) of the data set matrix, in step 3a the method performs a standardization of matrix $\boldsymbol{M}$ and creates a new matrix $\boldsymbol{T}$ with the same dimension as matrix $\boldsymbol{M}$, that is, both matrices are $(N \times F)$, where N is the number of samples in the training set and F is the number of features. Now, the $t_{s,f}$ cell of $\boldsymbol{T}$ contains the standardized value of feature f for music s, which is given by

$$t_{s,f} = \frac{m_{s,f} - m_{.,f}}{\sqrt{var(M_f)}} \quad , \tag{1}$$

where $m_{.,f}$ is the mean value of the fth column of matrix $\boldsymbol{M}$, that is,

$$m_{.,f} = \frac{1}{N}\sum_{i=1}^{N} m_{i,f} \ , \tag{2}$$

and the variance of feature $f, var(M_f)$, is obtained from

$$var(M_f) = \frac{1}{N-1}\sum_{i=1}^{N}(m_{i,f} - m_{.,f})^2. \tag{3}$$

As we will show in Sect. 5, depending on the combination of the initial groups of features, the number of columns of $\boldsymbol{M}$ and $\boldsymbol{T}$ may be more than 800. Thus, a strong feature reduction has to be made. So, At step 3b, a sample similarity matrix $\boldsymbol{S}$ is calculated:

$$\boldsymbol{S} = \begin{bmatrix} Sim(s_1,s_1) & Sim(s_1,s_2) & \dots & Sim(s_1,s_N) \\ Sim(s_2,s_1) & Sim(s_2,s_2) & \dots & Sim(s_2,s_N) \\ \vdots & \vdots & \ddots & \vdots \\ Sim(s_N,s_1) & Sim(s_N,s_2) & \dots & Sim(s_N,s_N) \end{bmatrix} \tag{4}$$

Each cell of the symmetric matrix $\boldsymbol{S}$ represents the similarity between two music samples and it is calculated by the following correlation from values of matrix $\boldsymbol{T}$:

$$Sim(s_i,s_j) = \frac{cov(s_i,s_j)}{\sqrt{cov(s_i,s_i)} \cdot \sqrt{cov(s_j,s_j)}} \ , \tag{5}$$

where the covariance between music samples s_i and s_j is given by

$$cov(s_i,s_j) = \frac{1}{F-1}\sum_{f=1}^{F}(t_{s_i,f} - t_{s_i,.})(t_{s_j,f} - t_{s_j,.}) \ , \tag{6}$$

where $t_{s_i,.}$ is the mean value of the ith line of $\boldsymbol{T}$.

Each line of matrix $\boldsymbol{S}$, corresponds to a music sample, now characterized by its similarity (within a range from -1 to +1) to all the other samples in the training set. On the other hand, each column of $\boldsymbol{S}$ may be seen as a new feature reflecting the similarity between a music sample and all the other samples. Clearly, there are as many columns as the number of music samples in the training set. Thus, with $\boldsymbol{S}$, the number of features is reduced from the number of initial attributes, usually very high, to a number which is equal to the size of the training set, which may be a much smaller number. As we will show in Sect. 5 we obtained good results using a training set of 60 samples.

Since samples of the same genre tend to show high similarities and, thus, there are strong correlations between the features in $\boldsymbol{S}$, another reduction in dimensionality can be obtained by a technique based on Principal Component Analysis (PCA) [1].

Since $\boldsymbol{S}$ is symmetric, it can be described as $\boldsymbol{S} = \boldsymbol{P\Lambda P^T}$, where $\boldsymbol{P} = [e_1, ..., e_N]$ is the orthogonal matrix of normalized eigenvectors of $\boldsymbol{S}$, and $\boldsymbol{\Lambda}$ is the diagonal matrix of its eigenvalues, $\lambda_1, ..., \lambda_N$, such that $\lambda_1 \geq ... \geq \lambda_N \geq 0$. Since $\boldsymbol{\Lambda}$ is symmetric, $\boldsymbol{\Lambda} = \boldsymbol{\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}}$ and $\boldsymbol{\Lambda^{\frac{1}{2}}} = \boldsymbol{(\Lambda^{\frac{1}{2}})^T}$. Thus,

$$\boldsymbol{S} = \boldsymbol{P\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}P^T} = \boldsymbol{P\Lambda^{\frac{1}{2}}}(\boldsymbol{\Lambda^{\frac{1}{2}}})^T\boldsymbol{P^T} = \boldsymbol{P\Lambda^{\frac{1}{2}}}(\boldsymbol{P\Lambda^{\frac{1}{2}}})^T = \boldsymbol{QQ^T}, \tag{7}$$

with

$$\boldsymbol{Q} = \boldsymbol{P\Lambda^{\frac{1}{2}}} \ . \tag{8}$$

The lines in matrix $\boldsymbol{Q}$ represent the music samples while the columns represent new uncorrelated features; see [1] for more details about this PCA-based technique. The leftmost columns of $\boldsymbol{Q}$ correspond to the most informative features. Thus, in order to reduce the number of features, we can discard the least informative ones, by ignoring the columns of $\boldsymbol{Q}$ having a variance (given by cells in $\boldsymbol{\Lambda}$) lower that a threshold which we set to 1. We call $\boldsymbol{R}$ to this new reduced matrix (step 3c) of Fig. 1, which is a copy of the k leftmost columns of $\boldsymbol{Q}$. This way we build a k-dimensional space in which music samples are represented: the k features correspond to the k axis in this new space, and matrix $\boldsymbol{R}$ contains the values for these features for each music in the training set. We tried other criteria associated with other threshold values, but this one provided a more reduced number of columns keeping good results. With this technique, we were able to drastically reduce the number of initial dimensions, that is, features (in $\boldsymbol{S}$) from 60 to 7 final dimensions (in $\boldsymbol{R}$) when we used the training set described in Sect. 5. Now, we are able to submit the resulting matrix $\boldsymbol{R}$ to the clustering stage.

3.2 The Clustering Stage

In the clustering stage (box 4 in Fig. 1) we use the Model-Based Clustering Analysis (MBCA) as proposed by Fraley and Raftery [2]. This approach uses no initial information about the number of clusters nor their shape or orientation. It represents the data by several possible models, which are characterized by different geometric properties. With this approach, data is represented by a mixture model where each element corresponds to a different cluster. Models with varying geometric properties are obtained through different Gaussian parameterizations and cross-cluster constraints. Partitions (clusters) are determined by the EM (expectation-maximization) algorithm for maximum likelihood, with initial agglomerative hierarchical clustering (see [2] for details). This clustering methodology is based on multivariate normal mixtures. So, the density function associated to cluster c has the form:

$$f_c(\boldsymbol{x_i}|\boldsymbol{\mu_c}, \boldsymbol{\Sigma_c}) = \frac{e^{(-\frac{1}{2}(\boldsymbol{x_i}-\boldsymbol{\mu_c})^T\boldsymbol{\Sigma_c^{-1}}(\boldsymbol{x_i}-\boldsymbol{\mu_c}))}}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma_c}|^{\frac{1}{2}}} \ , \tag{9}$$

where vector $\boldsymbol{x_i}$ represents an element that belongs to cluster c. Clusters are ellipsoidal and centered at the means $\boldsymbol{\mu_c}$. The covariance matrix $\boldsymbol{\Sigma_c}$ determines

the geometric characteristics of the cluster. This clustering methodology is based on the parameterization of the covariance matrix in terms of the eigenvalue decomposition in the form $\boldsymbol{\Sigma_c} = \boldsymbol{\lambda_c D_c A_c D_c^T}$, where $\boldsymbol{D_c}$ is the orthogonal matrix of eigenvalues, which determines the orientation of the axes. $\boldsymbol{A_c}$ is the diagonal matrix whose elements are proportional to the eigenvalues of $\boldsymbol{\Sigma_c}$ and which determines the shape of the ellipsoid. The volume of the ellipsoid is specified by scalar $\boldsymbol{\lambda_c}$. Characteristics (orientation, shape and volume) of distributions are estimated from the input data, and can be allowed to vary between clusters, or constrained to be the same for all clusters. Once all models are created, MBCA uses the Bayesian Information Criterion (BIC) to measure the evidence of clustering for each pair (*model, number of clusters*), and the larger the value of BIC, the stronger the evidence for the pair. So, by choosing the pair having the larger BIC, the most reliable model is automatically obtained, and then a vector indicating which music samples belong to which cluster is returned by this clustering approach. In other words, clusters are automatically formed in a k-dimensional space, according to data in matrix $\boldsymbol{R}$.

4 The Classification Method

Once the clusters are learned, the classification method can be used to classify new music samples (not included in the training set). Fig. 2 shows the steps of this method, which we describe below.

4.1 Representing New Music Samples in the k-dimensional Space Built in the Clustering Phase

Given a new (test) music sample s_t, the classification method starts by representing it with the same initial feature set as that used in the clustering method. As a result, the music sample is represented by an F-dimensional vector $\boldsymbol{m_{s_t}}$ that contains the feature values for music s_t, that is $\boldsymbol{m_{s_t}^T} = [m_{s_t,f_1}, ..., m_{s_t,f_F}]$ (step 2 in Fig. 2). Recall that matrix $\boldsymbol{M}$ (from Sect. 3.1) is a matrix whose lines are vectors of this form for the music samples in the training set.

Afterwards, vector $\boldsymbol{m_{s_t}}$ needs to be transformed into a new vector that represents music s_t in the k-dimensional space built in the clustering process. Firstly, $\boldsymbol{m_{s_t}}$ needs to be standardized (box 3a in Fig. 2). This transformation aims to set equal importance (scale) to each feature in vector $\boldsymbol{m_{s_t}}$. Despite $\boldsymbol{m_{s_t}}$ has only one value for each feature, this standardization will take into account the feature values of the music samples in the training set. Thus, the means and standard deviations calculated by the clustering method are used such that each cell of the new vector $\boldsymbol{t_{s_t}^T} = [t_{s_t,f_1}, ..., t_{s_t,f_F}]$ is given by an equation similar to (1):

$$t_{s_t,f_i} = \frac{m_{s_t,f_i} - m_{.,f_i}}{\sqrt{var(M_{f_i})}}, \tag{10}$$

where $m_{.,f_i}$ and $var(M_{f_i})$ result from (2) and (3) respectively.

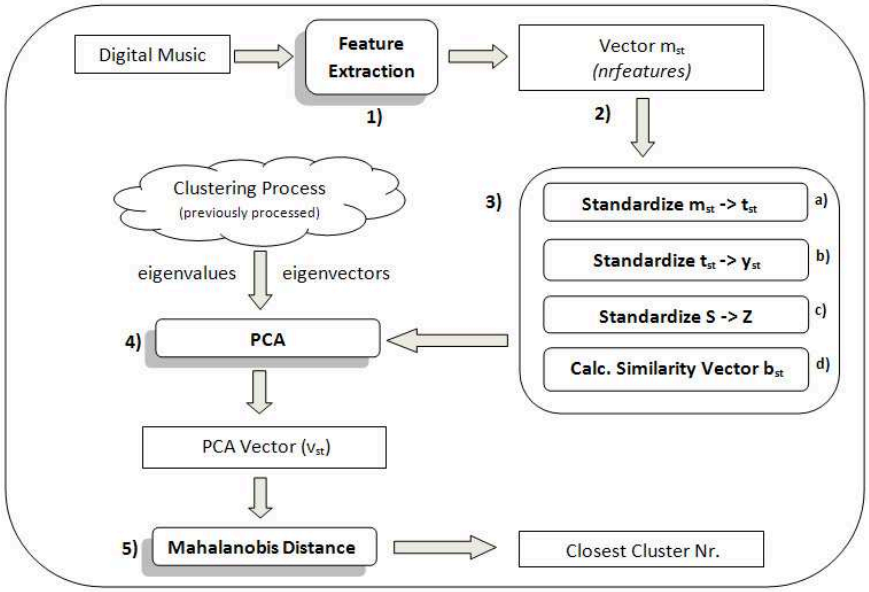


Fig. 2. The classification method

Now, the method aims to calculate a similarity vector between music s_t and all the samples in the training set. For that, we could use the correlation given by (5), used to calculate the similarity between the training set samples. However, since a correlation between non-standardized variables is equivalent to a covariance between the standardization of those variables, for reasons of computational weight, we followed this last option to get the same results. So, once we have vector $\boldsymbol{t_{s_t}}$, the referred standardization corresponds to a new vector $\boldsymbol{y_{s_t}^T} = [y_{s_t,f_1}, ..., y_{s_t,f_F}]$ (box 3b in Fig. 2), where y_{s_t,f_i} is the standardized value of feature f_i for music s_t, which is given by

$$y_{s_t,f_i} = \frac{t_{s_t,f_i} - t_{s_t,.}}{\sqrt{var(T_{s_t})}} \ . \tag{11}$$

$var(T_{s_t})$ stands for the variance associated to the t_{s_t,f_i} values for sound s_t along all features. So, $var(T_{s_t}) = \frac{1}{F-1}\sum_{f_i=1}^{F}(t_{s_t,f_i} - t_{s_t,.})^2$ and $t_{s_t,.} = \frac{1}{F}\sum_{f_i=1}^{F} t_{s_t,f_i}$.

At this step we need to relate our music s_t, now represented by $\boldsymbol{y_{s_t}}$, with the samples used in the learning process. In order to obtain a similarity vector $\boldsymbol{b_{s_t}}$ (box 3d in Fig.2), we need the information given by the similarity matrix $\boldsymbol{S}$, which may also be given by another matrix $\boldsymbol{Z}$ – see Appendix for details concerning matrices $\boldsymbol{S}$ and $\boldsymbol{Z}$ – such that each column of $\boldsymbol{Z}$ is a vector $\boldsymbol{z_s^T} = [z_{s,f_1}, ..., z_{s,f_F}]$ that represents the training set sample s using standardized values. In other words, each of these standardized values z_{s,f_i} is calculated

by

$$z_{s,f_i} = \frac{t_{s,f_i} - t_{s,.}}{\sqrt{var(T_s)}} \ . \tag{12}$$

Thus, vector $\boldsymbol{b_{s_t}}$ represents the similarity vector between $\boldsymbol{y_{s_t}}$ and each sample of the training set:

$$\boldsymbol{b_{s_t}^T} = \frac{1}{F-1}\boldsymbol{y_{s_t}^T Z} \ . \tag{13}$$

Now, by using the information obtained by the PCA-based technique from Sect. 3.1, that is, with $\boldsymbol{\Lambda}$ and $\boldsymbol{P}$, we can transform $\boldsymbol{b_{s_t}}$ into a vector $\boldsymbol{u_{s_t}}$, such that

$$\boldsymbol{u_{s_t}^T} = [u_{s_t,1}, ..., u_{s_t,N}] = \boldsymbol{b_{s_t}^T P \Lambda^{-\frac{1}{2}}} \ , \tag{14}$$

where N is still the number of samples of the training set.

Similarly to what was mentioned in Sect. 3.1 about the most informative columns of matrix $\boldsymbol{Q}$, only the k leftmost cells of $\boldsymbol{u_{s_t}^T}$ are used to obtain a final vector $\boldsymbol{v_{s_t}}$ that represents the music sample s_t in the k-dimensional space learned by the clustering method. In other words, $\boldsymbol{v_{s_t}} = [u_{s_t,1}, ..., u_{s_t,k}]$. In Appendix, the reader may see a detailed proof that $\boldsymbol{v_{s_t}}$ is the representation of music s_t in the k-dimensional space learned by the clustering method.

4.2 The Classification Stage

Now that music s_t is represented in the k-dimensional space learned by the clustering method, we need to relate $\boldsymbol{v_{s_t}}$ to the learned clusters (box 5 in Fig. 2). Mahalanobis distance was adopted for this purpose since it takes into account the geometric properties of each cluster, which is important since distances take different impact depending on the data dispersion along each axis. (This characteristic is not achieved when using other metrics such as Euclidean or Manhattan distances.)

The method calculates the Mahalanobis distance between each cluster centroid and $\boldsymbol{v_{s_t}}$, and proposes the class represented by the cluster having a smaller distance as the most likely class for music s_t. In other words, class c will be associated to $\boldsymbol{v_{s_t}}$ if $d(\boldsymbol{v_{s_t}}, \boldsymbol{\mu_c}, \boldsymbol{\Sigma_c^{-1}}) = \min_i d(\boldsymbol{v_{s_t}}, \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i^{-1}})$ where:

$$d(\boldsymbol{v_{s_t}}, \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}^{-1}) = (\boldsymbol{v_{s_t}} - \boldsymbol{\mu_i})^T \boldsymbol{\Sigma_i}^{-1} (\boldsymbol{v_{s_t}} - \boldsymbol{\mu_i}) \ , \tag{15}$$

where $\boldsymbol{\mu_i} = [\mu_{i.,1}, ..., \mu_{i.,k}]$ is the centroid of cluster i, with $\mu_{i.,f} = \frac{1}{\|\mathcal{C}_i\|} \sum_{s \in \mathcal{C}_i} r_{s,f}$. $\mathcal{C}_i$ is cluster i, that is, the set containing all the samples in this cluster, $\| \mathcal{C}_i \|$ is its size, and $r_{s,f}$ is the value of the fth axis (i.e., final feature) for music s (this is the value corresponding to the line associated to sample s and fth column of matrix $\boldsymbol{R}$, see Sect. 3.1). So, $\boldsymbol{\mu_i}$ represents an *average* music sample of cluster i. Finally, $\boldsymbol{\Sigma_i}$ reflects the geometric properties of cluster i in the k-dimensional space:

$$\boldsymbol{\Sigma}_i = \begin{bmatrix} E_{i_{1,1}} & E_{i_{1,2}} & \dots & E_{i_{1,k}} \\ E_{i_{2,1}} & E_{i_{2,2}} & \dots & E_{i_{2,k}} \\ \vdots & \vdots & \ddots & \vdots \\ E_{i_{k,1}} & E_{i_{k,2}} & \dots & E_{i_{k,k}} \end{bmatrix} \tag{16}$$

and

$$E_{i_{l,p}} = \frac{1}{\parallel \mathcal{C}_i \parallel -1} \sum_{s \in \mathcal{C}_i} (r_{s,l} - \mu_{i_{.,l}})(r_{s,p} - \mu_{i_{.,p}}) \ . \tag{17}$$

The most heavy calculations needed in the classification phase, such as the matrix $\boldsymbol{\Sigma_i^{-1}}$ for each cluster, can actually be made at the end of the clustering phase, as all needed data is available for that. This way, the classification of new music samples is a fast computation.

5 Results

In order to validate our approach, we used classical, metal, and reggae music samples from Tzanetakis' GTZAN [1] data collection [14], which has 100 music samples from several different genres. In addition, we added samples from a new genre, *fado*, and therefore, our data collection has 400 music samples (all represented with a sampling frequency of 22050 Hz, 16 bits, and single channel) representing 4 different music genres: classical, fado, metal, and reggae.

Table 1. Clustering error percentages for several feature combinations. The features are: timbral texture features (ttf), rhythm patterns (rssd_rh), beat, root mean square of the spectrogram frames (rmsFrame), MFCCs, spectral centroid + bandwidth + uniformity (centBandUnif), SSD over spectrogram (specStat), low-energy over sample spectrum (lener), spectral centroid (scentroid), and ZCR

	Feature Combination	Error (%)
1	'ttf','rssd_rh','beat','rmsFrame','mfccs','centBandUnif'	0
2	'ttf','rssd_rh','beat','rmsFrame','mfccs','specStat'	0
3	'ttf', 'rssd_rh', 'centBandUnif'	0
4	'ttf', 'rssd_rh', 'specStat'	2
5	'ttf','rssd_rh','beat','rmsFrame','mfccs','lener','scentroid', 'specStat'	3
6	'ttf','rssd_rh','beat','rmsFrame','mfccs','lener','scentroid','zcr', 'specStat'	3

Even though our clustering methodology does not use any information about the number of genres nor the genre of the samples, we used this labelling information to validate the results. Thus, once the clustering process is complete, we assume that each learned cluster c corresponds to the mostly represented genre in the cluster, and count the number of samples, o_c, in the cluster that have a different labeling. The overall error percentage is given by $e = (100 \sum_c o_c)/N$, where N is the number of samples.

[1] http://marsyas.info/download/

Table 2. Accuracy of the classification results for 3 different feature combinations

Feature Combination	Accuracy rate (%)
1	76.5
2	**81.8**
3	73.8

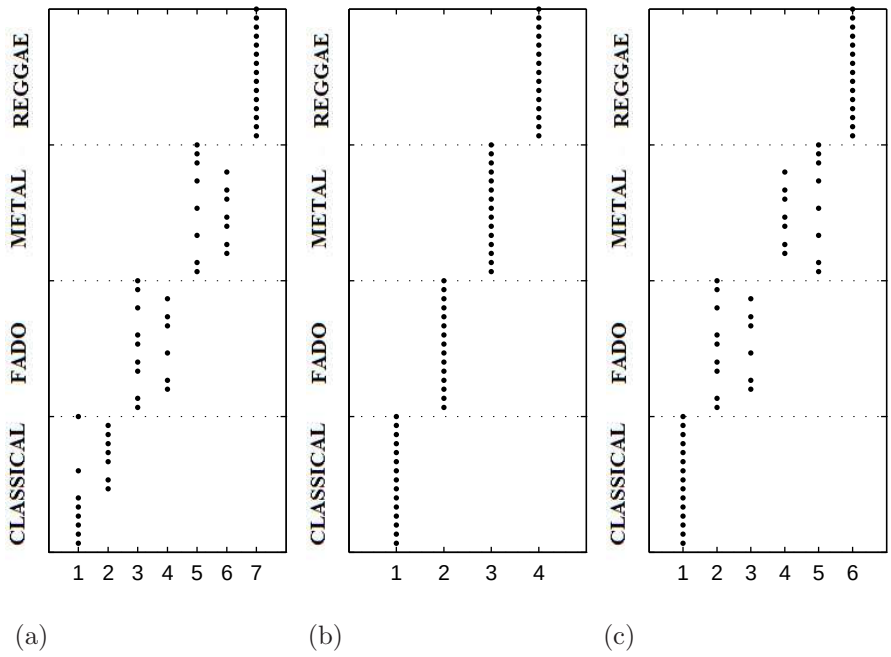
In order to validate the clustering methodology (described in Sect. 3), we used a training set composed of 60 elements (15 music samples from each of the four music genres mentioned above) and we tested many combinations of features (from Sect. 2). Table 1 shows the clustering results for the best feature combinations. The first three combinations have a 0% error rate, which shows that this approach is able to achieve perfect clustering results (assuming the initial labelling is correct).

Based on Table 1, it is clear that the third combination uses less features than the top two combinations. On the other hand, if we look at the clusters created (Fig. 3), the second combination achieves clustering results that perfectly match the initial labelling of the data. Nonetheless, this does not mean that combinations 1 and 3 achieve worse or incorrect results. It may actually be the case that these two combinations are learning sub-genres within classical, fado and metal. Each feature in the second combination actually represents a group of features as the whole number of real audio sub-features this combination represents is equal to 873.

Once the clusters were learned, we proceeded and classified new music samples. In order to evaluate the performance of the classification method (described in Sect. 4), we used a test set with the remaining 85 music samples (not used for clustering) for each of the four genres, making a total of 340 samples. Tests were made for clusters learned from each of the first three combinations from Table 1. As can be seen in Table 2, combination 2 achieves the best accuracy (precision) results with 81.8% correctly classified samples, which is a very satisfactory result, given this is an unsupervised approach.

6 Conclusions and Future Work

We proposed an unsupervised clustering and classification methodology for automatic genre classification. This kind of approach has the advantage of being totally independent of any influence from a human taxonomy. Since music genres do not present clear boundaries between them, and human genre taxonomy is hard to be achieved, we believe that an unsupervised approach is more suitable for music genre classification, while a supervised approach based on previously labelled data tends to be subjective. Besides, by learning directly from data in features, an unsupervised approach may automatically detect new genres, which is not possible for the more static nature of the supervised approaches.



(a) (b) (c)

Fig. 3. Clustering results for combinations (a) 1, (b) 2 and (c) 3 from table 1. The x-axis shows the clusters learned, while the y-axis shows the initial labeling of the data. For instance, figure (a) shows that 8 classical music samples fell in cluster number 1 while 7 classical samples fell in cluster number 2

In order to learn the clusters, the clustering method uses only the audio features of the training samples, and no previous knowledge on the genre of the individual samples. In addition, no information on the number of clusters is given a priori. This method achieves a perfect clustering accuracy (100%) when tested with four music genres (even though the genre labeling was not used in the training process, the results agreed with the manual labelling of the data), which shows that it is possible to achieve good accuracy results using an unsupervised method. In addition, as discussed in section 5, depending on the audio features used, the method is also able to find sub-genres within the data.

Once the clusters are learned, the classification method can categorize new music samples according to the previously learned clusters. This method uses Mahalanobis distance so that it is able to deal with clusters of different shapes, volumes and orientations. An 82% classification rate was obtained with four music genres.

We noticed that some misclassified samples were almost equidistant (in terms of Mahalanobis distance) to the chosen cluster and their actual cluster. This suggests that, as future work, a further analysis must be done to detect the existence of possible patterns of the Mahalanobis distances to every cluster,

for both cases (correct classifications and incorrect ones). Such analysis may be important, for instance, to decide if a sample that is too distant from all clusters must be rejected; or to suggest that new clusters ought to be learned (which should be done by running the clustering mehtod again) because several samples are approximately equidistant to two given clusters.

Even though we only reported the clustering and classification results for four music genres, we are currently investigating how the system behaves with more genres. Although this work is still not finished, we were already able to confirm that this clustering method can achieve good results with five and six music genres (at least around 90% clustering accuracy). Working with more music genres may require the use of more (or different) audio features. There are other audio features that we did not explore yet but could be important to discriminate other genres.

Finally, in order to test different feature combinations we simply used a brute force method, that is, with no prior selection. Instead, a possible *filtering* over the extracted features should also be explored in future work, as to only process those features that present higher variances between music samples.

Appendix

Here we prove that the test samples are represented in the k-dimensional space learned by the clustering method, that is, that $\boldsymbol{v_{s_t}}$ (see Sect. 4.1) is the translation of test music sample s_t in this k-dimensional space.

Proof. Let us suppose we want to classify a sound, say the first music of the training set, which is available in $\boldsymbol{z_1}$, the first column of matrix $\boldsymbol{Z}$. So, by (13) $\boldsymbol{b_1}^T = \frac{1}{F-1}\boldsymbol{z_1}^T\boldsymbol{Z}$ since now $\boldsymbol{y_{s_t}}$ is substituted by $\boldsymbol{z_1}$; $\boldsymbol{z_1}^T = [z_{1,1}, \ldots, z_{1,F}]$. Then $\boldsymbol{b_1}^T = [b_{1,1}, \ldots, b_{1,N}]$ where

$$b_{1,j} = \frac{1}{F-1}\sum_{i=1}^{F} z_{1,i} \cdot z_{j,i} \ . \tag{18}$$

Notice that, by statistics theory, (18) and (5) give the same result since $Sim(s_i, s_j)$ in (5) is a correlation using non-standardized values, and $b_{i,j}$ in (18) (generalizing from 1 to i) is a covariance using the standardization of those values. Then, in order to simplify this proof, let us suppose that we want to classify not just one music from the training set, but the whole training set. Then it is easy to conclude that

$$\boldsymbol{B} = \frac{1}{F-1}\boldsymbol{Z^T Z} \ . \tag{19}$$

$\boldsymbol{B}$ would be obtained instead of $\boldsymbol{b_1}$. Note that $\boldsymbol{B} = \boldsymbol{S}$, being $\boldsymbol{S}$ the similarity matrix given by (4), because it contains the similarity vectors between each training set music and all other music samples.

Now let us work with the entire $\boldsymbol{S}$ as if we wanted to translate all training sounds into vectors in the k-dimensional space. Then, from (14) we would obtain

$\boldsymbol{G} = \boldsymbol{SP\Lambda^{-\frac{1}{2}}} = \boldsymbol{SP\Lambda^{-1}\Lambda^{\frac{1}{2}}}$, but since $\boldsymbol{P^TP} = \boldsymbol{I}$ (where $\boldsymbol{I}$ is the identity matrix) and $\boldsymbol{S} = \boldsymbol{P\Lambda P^T}$, then $\boldsymbol{G} = \boldsymbol{P\Lambda P^T P\Lambda^{-1}\Lambda^{\frac{1}{2}}} = \boldsymbol{P\Lambda\Lambda^{-1}\Lambda^{\frac{1}{2}}} = \boldsymbol{P\Lambda^{\frac{1}{2}}}$.

But $\boldsymbol{P\Lambda^{\frac{1}{2}}} = \boldsymbol{Q}$, the matrix characterizing all sounds by the PCA-based method presented before (see (8)). Since we used a *copy* of the whole training set for classification instead of just one music, we obtained a matrix ($\boldsymbol{Q}$) instead of a vector $\boldsymbol{u_1}$. Then, choosing the k leftmost columns of this matrix we would obtain matrix $\boldsymbol{R}$ referred in Sect. 3.1, which contains the representation of the whole training set in the k-dimensional space. With this, we proved that $\boldsymbol{v_{s_t}}$ is the representation of the test music sample in the k-dimensional space learned by the clustering method.

Acknowledgments

This work was part of the Videoflow project and partially funded by *Quadro de Referência Estratégica Nacional* (QREN) and *Fundo Europeu para o Desenvolvimento Regional* and *Programa POR Lisboa*.

Bibliography

[1] Y. Escoufier and H. L'Hermier. A propos de la comparaison graphique des matrices de variance. *Biometrical Journal*, 20(5):477–483, 1978.

[2] C. Fraley and A. E. Raftery. How many clusters? which clustering method? answers via Model-Based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[3] S. Golub. Classifying recorded music. Master's thesis, University of Edinburgh - Division of Informatics, 2000.

[4] A.L. Koerich and C. Poitevin. Combination of homogeneous classifiers for musical genre classification. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 1, pages 554–559, 2005.

[5] C. Lee, J. Shih, K. Yu, and H. Lin. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *IEEE Transactions on Multimedia*, 11(4):670–682, 2009.

[6] T. Li and M. Ogihara. Toward intelligent music information retrieval. *Multimedia, IEEE Transactions on*, 8(3):564–574, 2006.

[7] T. Lidy and A. Rauber. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. In *ISMIR*, pages 34–41, 2005.

[8] M.F. Mckinney, J. Breebaart, and Prof Holstlaan. Features for audio and music classification. In *ISMIR*, 2003.

[9] D. Pye. Content-based methods for the management of digital music. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 6, pages 2437–2440 vol.4, 2000.

[10] A. Rauber, E. Pampalk, and D. Merkl. Using Psycho-Acoustic models and Self-Organizing maps to create hierarchical structuring of music by sound similarity. In *ISMIR*, 2002.

[11] X. Shao, C. Xu, and M.S. Kankanhalli. Unsupervised classification of music genre using hidden markov model. In *IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, pages 2023–2026, 2004.

[12] H. Soltau, T. Schultz, M. Westphal, and A. Waibel. Recognition of music types. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 1137–1140, 1998.

[13] T. Tolonen and M. Karjalainen. A computationally efficient multipitch analysis model. *IEEE transactions on speech and audio processing*, 8(6):708–716, 2000.

[14] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE Transactions on*, 10(5):293–302, 2002.

[15] G. Tzanetakis, A. Ermolinskyi, and P. Cook. Pitch histograms in audio and symbolic music information retrieval. *Proceedings of the third international conference on music information retrieval (ISMIR)*, pages 31–38, 2002.