A Music Classification Model based on Metric Learning and Feature Extraction from MP3 Audio Files

Angelo C. Mendes da Silva[a], Maurício A. Nunes[b], Raul Fonseca Neto[a,*]

[a]*Departament of Science Computer, Universidade Federal de Juiz de Fora, Brazil*
[b]*Departament of Science Computer, IFSudesteMG, Brazil*

Abstract

The development of models for learning music similarity and feature extraction from audio media files is an increasingly important task for the entertainment industry. This work proposes a novel music classification model based on metric learning and feature extraction from MP3 audio files. The metric learning process considers the learning of a set of parameterized distances employing a structured prediction approach from a set of MP3 audio files containing several music genres. The main objective of this work is to make possible learning a personalized metric for each customer. To extract the acoustic information we use the Mel-Frequency Cepstral Coefficient (MFCC) and make a dimensionality reduction with the use of Principal Components Analysis. We attest the model validity performing a set of experiments and comparing the training and testing results with baseline algorithms, such as K-means and Soft Margin Linear Support Vector Machine (SVM). Experiments show promising results and encourage the future development of an online version of the learning model.

Keywords:
music similarity, metric learning, feature extraction, mel frequency cepstral coefficient, principal components analysis

1. Introduction

Knowledge about a customer's preference or user's profile allows the opportunity to offers products in a personalized way and, consequently, can directly influence the service or purchase of a particular product. In the millionaire market of music streaming platforms, learning the customer's preference is crucial for maintaining its fidelity and help the acquisition of new subscribers. Among their tactics to retain the customer, these platforms frequently use techniques as recommendation systems offering single songs or in the form of playlists. In general, this recommendation is based on expert-added tags (Barrington et al., (2009) or collaborative filters (McFee et al., (2010).

In this work, it is sought to measure the similarity between musics using acoustics characteristics, to remove the subjectivity of the tags attributed to the music samples and to avoid the imbalance between the

*Corresponding author
Email addresses: angelo.mendes@ice.ufjf.br (Angelo C. Mendes da Silva), archanjomau@yahoo.com.br (Maurício A. Nunes), raulfonseca.neto@ufjf.edu.br (Raul Fonseca Neto)

number of users and the evaluations contained in a database. In this sense, we propose a novel approach to learning the customer's preference based on the study of music similarity using a metric learning approach. Also we opted to extract the music information directly from the MP3 audio files.

We consider for each sample the use of a thirty seconds long audio segment. We then extract an audio feature vector from the musical segment using the Mel-Frequency Cepstral Coefficient (Loughran et al., (2008) with the objective to capture the audio signal. Due to the large number of extracted features we made a study of dimensionality reduction using the Principal Components Analysis (PCA) instead of a Feature Selection approach. This feature representation approach based only on MFCC features is more reduced and homogeneous compared to others works. For example, in (Wolff and Weyde, (2014) it is used a set of low level (chroma and timbre vectors) and high level (loudness, beat and tatum means and variances) features, (McKinney and Breebaart, (2003) that made a comparative study involving four groups of audio features (low-level signal, MFCC, psychoacoustic and auditory model) and in (Bergstra et al., (2006) that extracted and includes a set of audio features from different methods of audio signal processing such as MFCC, Fast Fourier Transform (FFT), Real Cepstral Coefficients (RCEPS) and Zero-crossing Rate (ZCR).

The Metric Learning problem has been solved as an optimization problem and considers the minimization of a set of parameterized distances measured over pairs of samples and subject to triangular inequality constraints (Xing et al., (2002) . Also the distance values must be symmetrical and non-negatives. In this context, different solutions are verified, such as learning a full parameter matrix or a diagonal matrix, resulting in a parameters vector. In this latter case we learn a metric that weighs the different dimensions of the problem space. This approach can be considered as the use of a contrastive loss (Hadsell et al., (2006) that tries to minimize a parameterized distance between similar samples and to maximize between those dissimilar.

The proposed method for learning the music similarity has a direct relationship with the Structured Prediction Problem (Coelho et al., (2017). It is based on the fulfillment of a set of constraints that attest the pertinence of each music sample in relation to it respective genre centroid when compared to other alternatives. These constraints represent the condition that the parameterized distance between a sample and it respective centroid must be smaller than to any other centroid of the training set. The work developed by Wolff and Weyde (Wolff and Weyde, (2014) also uses an analogous approach for learning the music similarity but, in this case, the authors consider the learning of a distance metric from relative comparisons (Schultz and Joachims, (2004) involving for each constraint a triple of audio samples.

We provide an extensive evaluation of the model's performance by making a set of training and testing

experiments. We compare the results obtained with the model against a multiclass classifier based on a soft margin linear SVM trained with an one-against-all strategy. We also study the influence of the metric learning, with variations on segment length, feature dimensionality and in the training set size and their respective impacts in the generalization success. Our experiments show that the metric learning from comparisons to genre centroids has a positive effect in the process of music similarity learning. In the experiments we use two types of datasets. The first is the public GTZAN dataset that consists of 1000 audio segments with 30 seconds each, equally partitioned in 10 genres, and the second is an artificial music dataset which reports a customer's music preference also containing 1000 audio segments with 30 seconds each but with 5 genres only. Both datasets were constructed a variation of audio segments containing 15 seconds. In addition, the dataset MUSIC generated two more variations with number of audio segments equal to 250 and 500 audio segments containing 30 seconds.

In addition to this introduction the remainder of this work is organized as follow: Section 2 reports on related work. Section 3 reports the process of feature extraction from MP3 media audio files. Next, in section 4, we present the process of learning music similarity. We report our experiments and the discussion of results in section 5. Finally, section 6 presents the conclusions and perspectives of future work.

2. Related Work

Many areas of research in music information retrieval involve classification tasks like music recognition, genre classification, playlist generation, audio to symbolic transcription, etc. The fundamental information that supports music classification includes musical data collections (called instances), audio recordings, scores and cultural data (playlists, album reviews, billboard stats, etc.) which also can include meta-data about the instances like artist identification, title, composer, performer, genre, date, etc. This musical data collection is very complex and in our approach, can be resumed by a feature extraction process, wherein features represent characteristic information about music instances. Additionally, Machine Learning (ML) algorithms can learn how to associate feature vectors of instances with their classes for music classification (Gupta, (2014).

In (Vlegels and Lievens, (2017) is reported the attempt to discover clusters that represent the similarity relation between musics from a set of artists and from user's profile information obtained from different locations and distributions of genre and age. These informations are extracted from answers to questions about socio-demographic aspects and their cultural behavior in a broad range of domains like arts, everyday culture, leisure activities, sport, and recreation. The major objective was to learn the customer's musical

preference using the relationship between his profile and information about his favorite artists. For this, the authors explore the music knowledge provided in social network to construct a two-mode network of people and music artists. The results show that research using information based only on cultural information and genre preferences might be inadequate or insufficient for a better classification because it misses important discriminate information that cannot be captured.

The high complexity to evaluate the music similarity is reported in (McFee et al., (2010), which describes the need to incorporate acoustic, psychoacoustic and theoretical characteristics derived from audio information to obtain better classification results. Therefore, similarity between musics plays a central role in music recovery, recommendation and classification tasks. Observe that from a reference music we can use a query to return several others with similar characteristics, indicating new preferences and also label unknown samples based on similarity metrics (Pampalket, (2006)(Wolff and Weyde, (2014)(Slaney et al., (2008). In (Wolff and Weyde, (2014) the author's conclusion shows that the importance of each feature of a given music and its similarity measure with others is highly dependent on the context in which the music is inserted. In this sense, it is noticed the importance that learning models have when helping to ensure that a recommendation system is appropriate for each customer's preference.

Several approaches that use music similarity analysis to perform a task have a common characteristic in which the user's feedback is ignored and the systems adopt a common sense on the perception of music similarity (Barrington et al., (2009). For example, a band will always sing a song of only one genre, or songs from a region will always be inserted into the same group, due to cultural influence and other factors that are nontransparent to the user (McFee et al., (2010). However, these approaches cannot work well with an online music learning model building from user's feedback.

A technique commonly used to assess similarity between musics with user's feedback information is the Collaborative Filtering (McFee et al., (2010) (Gupta, (2014). Collaborative Filtering aims to individualize the user's profile based on the evaluations they perform in the system. However, this technique presents several problems (Herrada, (2010) such as sparsity due to non-existent evaluations in the base, subjectivity since users can differ in evaluations on the same data and scalability because the complexity tend to increase proportionally in relation to the number of evaluations. Thereby, the major difficulty in evaluating music similarity based on information coming from collaborative filters or metadata is the existence of a large number of uncertain data and noises that leads to a large incoherence in the evaluation process and consequently in the performance of the system. As a solution to these problems, we have to use audio content information with the intention of removing the subjectivity. In this way, the feature vector extracted

from audio information of all instances will be comparatively analyzed with the same criteria improving the overall performance (Slaney et al., (2008). In this sense, a better perspective is to use information obtained from audio content and user's preferences without limiting itself to metadata used for music description. It is expected that audio content allows us to learn the user's preference in a more objective way analyzing the music structural composition and its features information.

The method for music similarity learning proposed here is an extension of the work presented in (Coelho et al., (2017), in which the authors developed an approach directly related to the Structured Prediction Problem. It is based on the fulfillment of a set of pairwise comparison constraints. These constraints scale in order O(n) with the number of samples and represent the condition that the parameterized distance between a music and its respective genre centroid must be smaller than in relation to any other alternative. Also, we use a margin-based contrastive loss function ensuring that musically similar examples are embedded together with this respective genre clusters. As previously mentioned, our work has a great similarity with the model of relative comparisons proposed in (Wolff and Weyde, (2014) that have a structured SVM approach (Schultz and Joachims, (2004). In this model, each constraint represents the similarity relation between a triple of samples reflecting the fact that a sample x_i is more similar to sample x_j than to sample x_k. The major drawback of this formulation is the number of constraints that scale in order $O(n^3)$ with the number of samples.

Following this ML approach in (Bergstra et al., (2006) the authors proposed a learning algorithm based on a multiclass version of an ensemble leaner ADABOOST (Schapire and Singer, (1999). The authors made a comparative study of your algorithm with other ML techniques like SVM and Artificial Neural Networks. It's important to highlight that the performance of SVM is better when only MFCC features are used and the segments length is about thirty seconds. In (McKinney and Breebaart, (2003) the classification of audio files was performed using a quadratic discriminant analysis (Duda and Hart, (1973). The model uses a n-dimensional Gaussian Mixture and, consequently, each genre has its own mean and variance parameters. The authors also made a comparative study of feature representation and the MFCC features produced better results for classification.

3. Feature Extraction

3.1. Mel Frequency Cepstral Coefficient

The work of (McKinney and Breebaart, (2003) carried out a study on the impact that temporal and static behaviors of a set of features can have on the performance of classification of general audios and genres of

music. The major conclusion of this work is that the variation of the accuracy in the classification process is conditioned by the choice of the feature vector that best represents the audio data set, independently of the computational resource used to construct them. Among the features sets analyzed, two presented higher performance to the classifier, Auditory Filterbank Temporal Envelopes (AFTE) and Mel Frequency Cepstral Coefficient (MFCC). Also, the works of (Bergstra et al., (2006), (Burred and Lerch, (2004) and (Yen et al., (2014) highlight the use of MFCC of an audio signal to construct the analogue feature vector in music classification tasks.

According to (McKinney and Breebaart, (2003) we describe the whole feature set used to represent audio signals that obtained the best classification results. The first feature set, AFTE, is a representation model of temporal envelope processing by the human auditory system. Each audio frame is processed in two stages: (1) it is passed through a bank of 18 4th-order bandpass filters spaced logarithmically from 26 to 9795 Hz; and (2) the modulation spectrum of the temporal envelope is calculated for each filter output. The spectrum of each filter is then summarized by summing the energy in four bands.

The Table 1 presents the feature vector extracted from AFTE with its 62 features:

Table 1. AFTE Feature

Interval of Features	**Description**
1-18	DC envelope values of filters 1-18
19-36	3-15 Hz envelope modulation energy of filters 1-18
37-52	20-150 Hz envelope modulation energy of filters 3-18
53-62	150-1000 Hz envelope modulation energy of filters 9-18

The second feature set is based on the first 13 MFCCs. The Table 2 presents the final feature vector with its 52 features:

Comparing to the other two sets of features of the work, low level features and Psychoacoustics, the MFCC set presented better results to classify both general audios and music and also has a lower dimensionality and less complexity in the extraction process. However, when the AFTE set is used there is an improvement in the classification results although it is not considered to be statistically significant. Due for this consideration we opted to use only the MFCC technique for feature extraction purposes. MFCC is a standard preprocessing technique in speech processing. They were originally developed for automatic speech recognition (Oppenheim, (1969), and have proven to be useful for music information retrieval, classification

Table 2. MFCC Feature

Interval of Features	Description
1-13	DC values of the MFCC coefficients
14-26	1-2 Hz modulation energy of the MFCC coefficients
27-39	3-15 Hz modulation energy of the MFCC coefficients
40-52	20-43 modulation energy of the MFCC coefficients

and many other tasks (Pampalket, (2006).

The MFCC attribute extraction technique performs an analysis of short-time spectral features based on the use of converted sound spectrum for a frequency scale called MEL (Stevens et al., (1937). It aims to mimic the unique characteristics perceptible by the human ear. These coefficients are a representation defined as the cepstrum of a timeshifted signal, which has been derived from the application of the Discrete Fourier Transform (DFT), in non-linear frequency scales (Siqueira, (2012).

According to (Yen et al., (2014) the MFCC extraction algorithm used by us is described as follows:

1) Pre-emphasis the audio signal with a pre-emphasis filter, magnify the high frequency part. By doing so, make the channel characteristics more clearly. The pre-emphasis filter is determined as:

$$y(n) = x(n) - px(n-1),$$

where $x(n)$ is the waveform of the voice, p is the pre-emphasis coefficient.

2) To reduce the edge effect, apply Hamming window to each frame whose waveform have already attached pre-emphasis filter, calculate the amplitude of the vector by DFT.

3) Use Mel Filter Bank compress the Amplitude Spectrum. The filter bank consists of 33 triangular shaped band-pass filters. The focus here is to generate Mel-Frequency. The dimension of Mel is the horizontal axis which is used to reflect the human auditory characteristics, its unit is mel. The lower the frequency, the narrower the interval, the opposite is also true, the higher, the wider. The human ear in the subtle low frequency sound could feel the pitch different, but in the high frequency sound, its goes harder.

4) Transform the compressed values by Discrete Cosine Transform (DCT) to remove the correlation between the signals in different dimension, map the signal into low dimensional space.

5) Using the low dimensional composition of the obtained cepstrum as MFCC feature values.

3.2. Vector Quantization

The feature extraction process produces a multidimensional feature vector for every music frame. In this study we have considered only 13MFCCs and vector quantization is used to minimize the data of the extracted features. The vector quantization process was applied to the 13MFCCs of the music generating a codeword that represents each song. From here, every time we refer to the feature vector obtained by extracting the MFCC, we are referring to the quantized vector.

Vector quantization is a method usually applied in data compression. However, it also finds applications in the field of signals processing, classification and data extraction. In vector quantization, the objective is to represent a certain distribution of data using a number of prototypes significantly smaller than the number of data, thus quantizing the original data distribution.

In a formal way, the vector quantization process is defined by an operator, the vector quantizer. A vector quantizer, Q, of dimension k with size N is defined as the mapping of a set I of L vectors in space R^k, in a set C which has N vectors of output, where $L \gg N$, contained in the same space R^k(Carafini, (2015).

$$Q : I \rightarrow C$$

where, $I = \{x_0, x_1, ..., x_{L-1}\}$ with $x_l \in R^k$. And, $C = \{y_0, y_1, ..., y_{N-1}\}$ with $y_i \in R^k$, $\forall\ i \in J = \{0, 1, ..., N-1\}$. The set C is called codebook, and each vector that composes it, y_i , is the codeword.

One of the methods to obtain the codebook is the Linde-Buzo-Gray (LBG) algorithm, also known as Generalized-Lloyd's Algorithm (GLA)(Southard, (1992), since it is the vector generalization of the proposed scalar algorithm by Lloyd(Lloyd, (1982). The LBG algorithm iteratively adjusts an initial codebook so as to reduce the distortion measure until a local minimum, by some convergence criterion, is reached(Linde et al., (1980).

LBG algorithm is like a K-means clustering algorithm which takes a set of input vectors $S = \{x_i \in R^d | i = 1, 2, ..., n\}$ as input and generates a representative subset of vectors $C = \{c_j \in R^d | j = 1, 2, ..., K\}$ with a user specified $K \ll n$ as output according to the similarity measure. According to (Southard, (1992) the LBG Algorithm used by us is described as follows:

1) Input training vectors $S = \{x_i \in R^d | i = 1, 2, ..., n\}$.

2) Initiate a codebook $C = \{c_j \in R^d | j = 1, 2, ..., K\}$.

3) Set $D_0 = 0$ and let $k = 0$.

4) Classify the n training vectors into K clusters according to $x_i \in S_q$ if $\| x_i - c_q \|_p \leq \| x_i - c_j \|_p$ for $j \neq q$.

5) Update cluster centers c_j, $j = 1, 2, ..., K$ by $c_j = \frac{1}{|S_j|} \sum_{x_i \in S_j} x_i$.

6) Set $k \leftarrow k + 1$ and compute the distortion $D_k = \sum_{j=1}^{K} \sum_{x_i \in S_j} \| x_i - c_j \|_p$.

7) If $\frac{(D_k - 1 - D_k)}{D_k} > \varepsilon$ (a small number), repeat steps $4 \sim 6$.

8) Output the codebook $C = \{c_j \in R^d | j = 1, 2, ..., K\}$,

The flowchart in Figure 1 illustrates the complete process for building the feature vector.

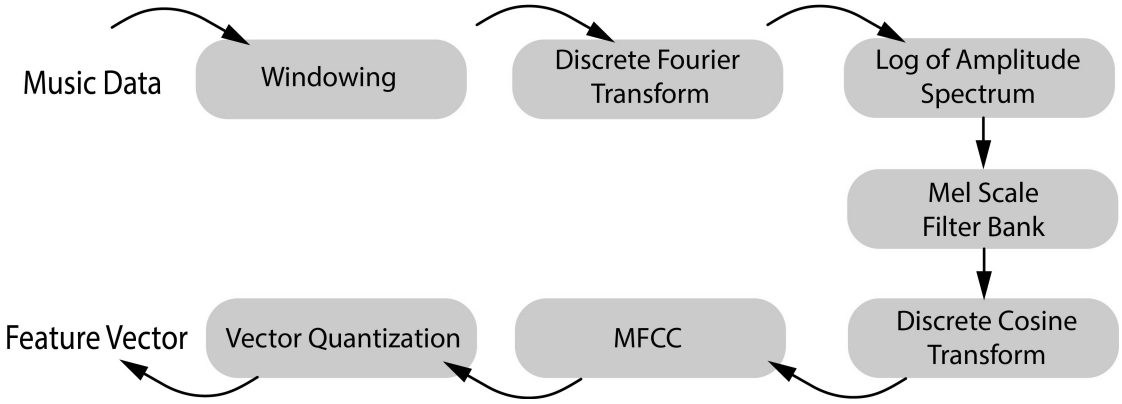

Figure 1: Feature vector building process

3.3. Dimensionality Reduction

Extracting MFCC feature values from audio segments makes the data volume extremely large, and different instances of duration can cause a variation in the dimensionality space. To reduce the cost of extracting and keeping all vectors with the same dimensionality, (Pampalket, (2006) showed that the information extracted over thirty seconds of a particular song is sufficient to represent it. Also, the main characteristics of the songs are exposed during their first half, and usually the second half is composed by the repetition of a considerable part of the first (Xin et al., (2014). In this sense, for feature extraction purposes, we build a MP3 audio segment with fifteen or thirty seconds of each music. Also, to improve the quality of the captured voice audio we made a jump of 15 seconds after the start. Statistically, much of this data is redundant and so we need to employ a method to extract the most significant information (Loughran et al., (2008). This is achieved through applying PCA.

PCA is a standard technique commonly used in statistical pattern recognition and in signal processing for performing dimensionality reduction. Essentially, it transforms data orthonormally so that the variance of the data remains constant, but is concentrated in the lower dimensions. The matrix of data being transformed consists of one vector of coefficients for each sample. Thus, there is now one matrix of data for each vector

of MFCC. The covariance matrix of the data matrix is then calculated. The principal components for the data set can be recovered from the eigenvectors of this covariance matrix. We defined empirically some dimension sizes that would be used in each of the datasets, with their variations, in order to evaluate the impact of dimensionality reduction.The maximum dimensionality of each dataset built after the PCA is limited to the number of music samples. For a music with 30 seconds, the feature vector obtained from the MFCC has a dimensionality equal to 1293 and for 15 seconds the size is 647. The process of creating each dataset was done with the feature extraction followed by the principal component analysis.

4. Learning from parametrized distances

4.1. Parameterized Distances and Similarity Relations

Let a set of n points in a d-dimensional space be defined as $\{x_i, i = 1, ..., n\} \subset R^d$. Also consider a set of constraints proposed by an expert pointing out the existence of a pairwise similarity set S that can be partitioned in k disjoints subsets: $S_1, S_2, ..., S_k$ each associated with a cluster. Therefore:

$$S : (x_i, x_j) \in S_l \rightarrow x_i \wedge x_j \text{are similar}$$

$$S = S_1 \cup S_2 \cup ... \cup S_k$$

Otherwise, if the points are dissimilar then we have:

$$D : (x_i, x_j) \in D_l \rightarrow x_i \wedge x_j \text{ are dissimilar}$$

Generally, the problem of metric learning with pairwise similarity relations is formulated as an optimization problem whose objective is to decrease the distances of similar pairs while increasing the distance with dissimilar ones. This approach involves a quadratic number of terms in objective function and a quadratic optimization problem. However, this problem can be reformulated as a simpler cluster analysis problem if we consider the existence of two graph properties:

Transitivity: if (x_i, x_j) and (x_j, x_k) are similar, then (x_i, x_k) are similar.

Symmetry: if (x_i, x_j) are similar, then (x_j, x_i) are similar.

Consider, also, a measure of parameterized distance between two points defined as a function of a matrix A_{dxd} positive semidefinite and symmetric (PSD):

$$d_A(x_i, x_j) = \| x_i - x_j \|_A^2 = (x_i - x_j)^T A (x_i - x_j), \quad (1)$$

with the following properties:

$$d_A(x_i, x_j) \geq 0,$$

$$d_A(x_i, x_j) = 0,$$

$$d_A(x_i, x_j) = d_A(x_j, x_i),$$

$$d_A(x_i, x_j) \leq d_A(x_i, x_k) + d_A(x_k, x_j)$$

In this sense, we can formulate the metric learning problem as a cluster analysis problem considering the relation of each subset S_l with a cluster. So, we have to solve:

$$Min \sum_l \sum_{(x_i,x_j)\in S_l} \| x_i - x_j \|_A^2 \tag{2}$$
$$subject\ to\ A \geq 0\ (PSD)$$

If we consider the case in which the parameters matrix is a diagonal matrix, we have to learn a vector of parameters $w = [w_1, w_2, ..., w_d]$, or an equivalent diagonal matrix W, whose solution is equivalent to rescaling the respective dataset of points. We can observe that if we consider the use of a identity matrix in (1) we have a set of Euclidean distances. Otherwise, if we adopted the covariance matrix then we have a set of Mahalanobis distances. For the more general case, we have a set of parameterized distances in function of a full matrix A. For the diagonal case the PSD constraint is fulfill if all components of vector w are non negatives, this is: $w_i \geq 0$. So, the formulation (2) for the diagonal case can be reformulated as:

$$\sum_l \sum_{(x_i,x_j)\in S_l} \| x_i - x_j \|_A^2 = \sum_l \eta_l \sum_{x_i \in S_l} \| x_i - c_l \|_A^2 =$$

$$= \sum_l \eta_l \sum_{x_i \in S_l} (x_i - c_l)^T A (x_i - c_l) =$$

$$= \sum_l \eta_l \sum_{x_i \in S_l} w_1(x_{i1} - c_{l1})^2 + w_2(x_{i2} - c_{l2})^2 + ... + w_d(x_{id} - c_{ld})^2, \tag{3}$$
$$subject\ to\ w_i \geq 0$$

This equivalence is proved by (Edwards and Cavalli-Sforza, (1965) taking into account the fact that each cluster centroid is computed as the mean of the pertinent cluster vectors, that is:

$$c_l = (\frac{1}{\eta_l}) \sum_i x_i, \forall x_i \in S_l$$

To the solution of the problem presented in (2) with a diagonal matrix (Xing et al., (2002) propose a relaxed formulation that involves the minimization of an unrestricted objective function with an additional penalty term:

$$Min \sum_{(x_i,x_j)\in S} \| x_i - x_j \|_A^2 - \log \sum_{(x_i,x_j)\in D} \| x_i - x_j \|_A$$

where S is the set of similar points and D is the set of dissimilar points.

Our approach to solve the metric learning problem is more close to the work of (Schultz and Joachims, (2004) that proposes an extension of Support Vector Machine (Cortes and Vapnik, (1995) and is based on the fulfillment of a set of comparative relation constraints. These comparative relations have the following expression involving a triple of points:

"x_i *is closer to* x_j *than to* x_k."

So, we can deduce that x_i is similar to x_j, but we cannot deduce with certainty that x_i are x_k similar or dissimilar. In this sense, becomes necessary to model a number of $O(n^3)$ constraints, where n is the total number of points, considering the representation of each subset of triples. Let w be the vector of parameters associated with each parameterized distance. Then, we can model each constraint as:

$$\forall i, j, k : d_w(x_i, x_k) - d_w(x_i, x_j) > 0. \tag{4}$$

This set of inequations can have innumerous solutions. In this sense, the authors proposed a solution similar to the flexible margin SVM considering the minimization of the Euclidean norm of the parameter vector w:

$$Min \frac{1}{2} \| w \|^2 + C. \sum\nolimits_{i,j,k} \xi_{i,j,k} \tag{5}$$

$$\text{subject to: } \forall i, j, k : d_w(x_i, x_k) - d_w(x_i, x_j) \geq 1 - \xi_{i,j,k}$$

$$\xi, w \geq 0$$

where ξ represents the vector of slack variables and C a penalty constant.

To overcome the drawback related to the high number of constraints we propose in our formulation a set of comparisons between each point and his respective cluster centroid reducing the number of constraints to $O(n)$. Also, as we shall see in the next subsection, we use as solution technique a relaxation method based on a structured version of perceptron model, thus avoiding the solution of a more complex quadratic programming problem.

4.2. Metric Learning with Structured Prediction

The Structured Prediction Problem is characterized by the existence of a training set $S = \{x(i), y(i), i = 1, ..., m\}$ formed by a collection of input and output pairs, where each pair is represented by a structured object $x(i)$ (input) and by a desired example $y(i)$ (output). The model aims to fulfill the constraints and correlations of the structured set of output Y relative to the input set X.

We can formulate the metric learning problem as a special case of the Structured Prediction model in which an input set X is formed by complete graphs and the output set Y is formed by subgraphs according to a set of similarity relations provided by an expert.

The inference problem can be solved as a minimization problem related to a function $S_x : Y(x) \rightarrow R$, that evaluates each particular output. Therefore, we should determine: $y* = arg\{min_{y \in Y(x)} S_x(y)\}$. This class of models can be parameterized by a vector w. Thus, considering: $w.f(x, y) = S_x(y)$, we have the following linear family of hypotheses:

$$H_w(x) = arg\{min_{y \in Y(x)}\{w.f(x, y)\}\}, \tag{6}$$

where $(x, y) \in S = \{x(i), y(i), i = 1, ..., m\}$, and the output y being subject to some constraint function $g(x, y)$. The goal is to estimate the vector w such that $Hw(x)$ maps any desired output y. Thereby:

$$y(i) \approx arg\{min_{y \in Y(x(i))}\{w.f(x, y)\}\}, \tag{7}$$

In this way, considering all output possibilities, we have:

$$\forall i, \forall y \in y(i) : w.f(x(i), y(i)) \leq w.f(x(i), y) \tag{8}$$

The solution of the structured prediction problem can be obtained by a maximal margin formulation according to (Taskar et al., (2005):

$$Min \frac{1}{2} \parallel w \parallel^2 \tag{9}$$
$$\text{subject to: } w.f_i(y_i) \leq min_{y \in Y(i)}\{w.f_i(y) + l_i(y)\}, \forall i,$$

where $f_i(y) = f(x(i), y)$ and the function $l_i(y)$ is defined as a loss function that scales the geometric margin value required for the false example y in relation to the selected example $y(i)$. If we consider only the fulfillment of the constraints this problem can be solved with the use of a variant of the Structured Perceptron algorithm (Coelho et al., (2012).

Now, the update rule without the loss function can be described as:

$$\text{for each pair } (x(i), y(i), i = 1, ..., m \text{ do}$$

$$\text{if } (w.f_i(y_i) > w.f_i(y^*)), \text{ then}$$

$$w \leftarrow w - \eta(f_i(y_i) - f_i(y^*)), \tag{10}$$

where $0 < \eta \leq 1$, is a constant learning rate and y^* the best candidate computed for each index i by an optimization algorithm.

Using an analogy with the update rule of the parameter vector associated with the metric learning problem, it can be said that $w.f_i(y_i)$ represents the value of the parameterized distance provided by the expert and $w.f_i(y^*)$ the value of the parameterized distance computed by the algorithm K-means. This distance function can be computed separately for each cluster considering the existence of m classes.

In this sense, the metric learning problem can be solved by computing the parameters vector w. Considering the fact that many solutions can fulfill all constraints it is also possible to adapt the structured prediction problem imposing a margin in order to find a unique vector solution. This is equivalent to minimize the Frobenius norm of the diagonal matrix W (Schultz and Joachims, (2004). Following (Coelho et al., (2012) we propose the following formulation:

$$Max\ \gamma \tag{11}$$

$$\text{subject to: } w.(f_i(y*) - f_i(y_i)) \geq \gamma. \parallel w \parallel, i = 1, ..., m$$

where γ is the margin parameter.

Now, the new update rule can be described as:

$$\text{for each pair } (x(i), y(i)), i = 1, ..., m,$$

$$\text{if } (w.f_i(y_i) > w.f_i(y*) - \gamma \parallel w \parallel), \text{ then}$$

$$w \leftarrow w(1 - \frac{\eta\gamma}{\parallel w \parallel}) - \eta(f_i(y_i) - f_i(y*) \tag{12}$$

The approach presented so far can be described as a batch correction process that considers the total intracluster error for each class where the vector w is updated by using the gradient method. However, considering the total error, the batch processing is responsible for large corrections in the w vector making

the gradient method unstable and requiring greater control of the learning rate. To overcome this problem, it is possible to consider the update rule for each individual error according to the labeling scheme provided by the expert. In other words, if the parameterized distance between a sample x_i and its respective centroid c_l is greater than the distance from the best candidate centroid c_k, where $k = arg\{min j \neq i \parallel x_i - c_j \parallel_w\}$ then we make the correction of the parameter vector w to force the fullfiment of this constraint. So, if we use the parameterized distance between two vectors, $dw(x_i, c_l) = (x_i - c_l)^T W(x_i - c_l)$, we have to solve the following margin maximization problem:

$$Min \frac{1}{2} \parallel w \parallel^2 + C. \sum\nolimits_i \xi_i \tag{13}$$

$$\text{subject to: } d_w(x_i, c_k) - d_w(x_i, c_l) \geq 1 - \xi_i, \forall i = 1, ..., n,$$

$$\xi, w \geq 0$$

where ξ represents the vector of slack variables and C the penalty constant.

In order to avoid the solution of a quadratic optimization problem, the margin maximization problem (13) can be solved as:

$$Max\ \gamma \tag{14}$$

$$\text{subject to: } d_w(x_i, c_k) - d_w(x_i, c_l) + \lambda\alpha_i \geq \gamma. \parallel w \parallel, \forall i = 1, ..., n,$$

$$\alpha, w \geq 0$$

where $\lambda = \frac{1}{C}$ represents the inverse of the penalty constant.

This formulation enable the soft margin relaxation process similar to the quadratic penalty of the vector ξ (Villela et al., (2016). Thus, the new update rule follows:

$$\text{for each pair } (x_i, c_l) \text{ do} \tag{15}$$

$$\text{if } d_w(x_i, c_k) - d_w(x_i, c_l) + \lambda\alpha_i < \gamma. \parallel w \parallel \text{ then}$$

$$w \leftarrow w(1 - \frac{\eta\gamma}{\parallel w \parallel} - \eta(d_w(x_i, c_k) - d_w(x_i, c_l)),$$

$$\alpha \leftarrow \alpha(1 - \frac{\eta\gamma}{\parallel w \parallel})$$

$$\alpha_i \leftarrow \alpha_i + \eta$$

The solution of problem (14) starts with a zero margin value. After the first execution of the structured perceptron with margin there is a greater possibility that the stop margin is not the maximum. This margin is considered as the margin with smaller value between the classes, thereby:

$$\gamma^t = min_{i=1,\ldots,m}\{\gamma_i\} \tag{16}$$

The new margin for a new iteration of the algorithm uses the double of the stop margin of the previous iteration, that is:

$$\gamma^{t+1} \leftarrow 2.\gamma^t \tag{17}$$

For the new problem we can use the final vector w of the previous iteration as initial solution. The stop margin is increased until the solution is not feasible. In this case, an approximation process based on a binary search can be used to find the maximum stop margin allowed.

For a label scheme predefined by an expert the problem (14) represents the inverse problem related to cluster analysis. That is, what should be an appropriate metric that fulfill the intracluster constraints? Otherwise, if the metric is predefined, the position of the centroids and consequently the scheme of labels will be computed using the same set of constraints based on distance comparisons.

4.3. The Parameterized Algorithms

The K-means algorithm minimizes the distance function known as intracluster related to a set of points distributed in the Euclidean space considering a number of clusters previously defined. More specifically, the algorithm minimizes the sum of the squares of Euclidean distances from each point to its respective centroid calculated as the average of their respectives points.

The parameterized distance function of the K-means algorithm is constructed based on the equivalence of the sum of the distances between vectors of the same cluster that share a similarity relation and the sum of intracluster distances. Thus, the only necessary change in the Euclidean K-means algorithm is in the determination of the center where now the parameterized Euclidean distances for the respective centroids must be used.

The Nearest Centroid Classifier (NCC) algorithm performs the comparison of the Euclidean distances of a new point to the respective class centroids, classifying the same according to the winner. On the other hand, the maximal Margin Parameterized Nearest Centroid (MMPNCC) uses a parameterized distance function for this purpose. If we consider a two-class classification problem with equal parameterized matrices we have as

classification hypothesis a linear decision function. Note that if we choose to learn two different parameters matrices, we have, as in the general case of a Fisher Discriminant, with two different covariance matrices, a quadratic decision function.

Indeed, (Fisher, (1936) proposes the first parametric algorithm for solving the problem related to classification in Pattern Recognition. For binary classification tasks with multivariate gaussian distribution respectively with centers m_1 and m_2 and covariance matrices Σ_1 e Σ_2 the decision function can be expressed according to the Bayes optimal solution as the output of the signal function:

$$f(x) = \varphi((x - m_1)^T \Sigma_1^{-1}(x - m_1) - (x - m_2)^T \Sigma_2^{-1}(x - m_2) + ln \mid \Sigma_2 \mid / \mid \Sigma_1 \mid) \tag{18}$$

According to (Cortes and Vapnik, (1995) the estimation of this function requires the determination of a quadratic number of parameters, that is, of order $O(d^2)$, where d is the dimension of the problem. However, when the number of observations is reduced compared to the number of parameters, lower than $10.d^2$, this estimate is no longer feasible. In this sense, Fisher in (Fisher, (1936) recommends the use of a linear discriminant function obtained from Eq. 18 when the covariance matrices are equal.

Let $w*$ be the optimal vector obtained from the metric learning process. Let W be the diagonal matrix that represents the components of $w*$. So, if we consider a two classes classification problem with a parameterized distance function with centroids m_1 and m_2 then we have the following linear decision function that represents the MMPNCC classifier :

$$f(x) = \varphi((x - m_1)^T W(x - m_1) - (x - m_2)^T W(x - m_2))$$

As will be seen in the next section, the proposed experiments aim to compare the use of the metric learning algorithm (MMP NCC), against state-of-art algorithm Support Vector Machine with Linear, Polynomial and Gaussian kernels, for music classification tasks.

5. Experiments and Results

5.1. Datasets

In this work we used two different datasets. One already known by several researches in the area of machine learning and the other one was constructed by the authors.

The former, named GTZAN[1], make possible to compare the accuracy of our results with significant works in the area of music learning. The latter, named MUSIC, aims to prove that the music similarity process with metric learning can be invariant with the training set or, in others words, with the customer's preference.

The GTZAN dataset appears in at least 100 published works, and is the most-used public dataset for evaluation in machine learning research of music genre classification (Sturm, (2013). The original dataset consists of 1000 audio segments each with 30 or 15 seconds length. It contains 10 genres (Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Popular, Reggae, and Rock), each represented by 100 segments.

The second dataset was divided into three nested subsets with respectively 250, 500 and 1000 audio segments. All subsets have the musics equally distributed in 5 genres (Rock, Classical, Jazz, Electronic and Samba) with each audio segment having 30 seconds that were extracted after the first 15 seconds of each music, as suggested by (Pampalket, (2006). For datasets with 1000 instances we also construct subsets with 15 seconds length.

From each segment the first 13 MFCCs were extracted, because this information is sufficient for discriminative analysis in the context of music classification tasks (Giannakopoulos and Pikrakis, (2014). Next, a vector quantization was made in order to produce a single feature vector. Finally we perform a principal component analysis in order to produce different dimensions of the feature vector and to represent the most statistically significant components.

In order to study of the impact of dimensionality reduction, we define empirically some dimension sizes that would be used in each of the subsets, with their variations. The maximum dimensionality of each subset built after the PCA is limited to the number of instances. For an audio segment with 30 seconds, the feature vector obtained from the MFCC has a dimension with size equal to 1293 and for 15 seconds the size is 647.

From the GTZAN dataset, subset variations were generated for studies on the impact of dimensionality reduction and audio segment length on a classification process. This dataset generated five subsets containing 1000 musics with 30 seconds and 50, 100, 250, 500 and 1000 dimensions and also four subsets with 1000 musics with 15 seconds and 50, 100, 250 and 500 dimensions.

In a process similar to what was done in the GTZAN dataset, variations on dimensionality and in the audio segment length in MUSIC dataset have been made in order to produce the desirable subsets. Additionally, to perform an analysis of the metric learning process in the training, we produce new subset variations

[1] http://marsyasweb.appspot.com/downloads/data_sets

with only 250 and 500 musics.

A summary of the two datasets is shown in Tables 1 and 2 with the number of samples, the time of each sample, the dimensions of each subset, and the number of classes.

Table 3. GTZAN dataset

samples	**seconds**	**dimensions**	**classes**
1000	15	50 - 100 - 250 - 500	10
1000	30	50 - 100 - 250 - 500- 1000	10

Table 4. MUSIC dataset

samples	**seconds**	**dimensions**	**classes**
250	30	50 - 100 - 250	5
500	30	50 - 100 - 250 - 500	5
1000	15	50 - 100 - 250 - 500	5
1000	30	50 - 100 - 250 - 500- 1000	5

5.2. Results

The first, Table 3, scenario used the datasets music250 and music500 both with MFCC extracted over 30 seconds of audio for the purpose of evaluating the metric learning process in the context of music similarity. Also we make a study of dimensionality reduction varying the feature vector dimension. We compare the results obtained by the parameterized algorithm against the classical K-Means algorithm employed together with the Nearest Centroid Classifier.

The result of this analysis is fundamental to demonstrate the ability of learning music silmilarity and to make possible the evolution of the classifier with a batch processing for an online classifier with the capacity to receive music in real time and to adapt the algorithm's parameters according to the customer's rating.

The second scenario, Table 4, used the datasets music250 and music500 both with MFCC extracted over 30 seconds and music1000 and GTZAN both with MFCC extracted over 15 or 30 seconds containing all variations related to the study of dimensionality reduction. We compare the results obtained by the parameterized algorithm against the classical SVM algorithm with soft margin using Linear, Polynomial and Gaussian kernels.

For a statistical analysis, all experiments were performed with 20 runs for each dataset in all its variations. The folders were selected randomly in a balanced way, 50% of the data for the training set and the other 50% of the data for the test set and use an one-against-one strategy to construct the multi-classification decision function. For SVM and Metric Learning algorithms the penalty constant C varied between 0.1 and 1.5 and the reported values are computed as an average. The results are presented in tables representing the mean values for the percentage of accuracy of the models and the variance obtained in each one of the experiments.

Tables 5 and 6 present the results of the training performance represented by the mean and variance computed over 20 runs. Notice that the classification error is considered when the difference of the respective distances relative to the correct genre centroid have negative value.

Table 5. Training performance to MUSIC dataset with 250 musics

	Euclidean K-means		**MMP K-means**	
dimension	μ	σ^2	μ	σ^2
50	32,52	5.461	55,00	6.514
100	37,14	5.002	66,14	5.437
250	34,30	4.907	60,10	3.264

Table 6. Training performance to MUSIC dataset with 500 musics

	Euclidean K-means		**MMP K-means**	
dimension	μ	σ^2	μ	σ^2
50	37,60	5.676	41,20	6.902
100	35,60	4.283	51,60	5.311
250	34,00	4.564	62,00	2.880
500	34,20	5.112	63,60	4.935

Observing the results reported in tables 5 and 6, we can assert that the metric learning approach is very important to learning music similarity. The baseline algorithm K-means with Euclidean distances has the effect of underfitting and consequently can not learn a correct decision function. Also, we can observe that the metric learning algorithm does not present the effect of overfitting.

Tables 7 and 8 present the classification results obtained respectively by the datasets music250 and

music500 compared with the SVM algorithm with the three types of kernel. Considering the variation on the dimensionality of the problem, the parameterized algorithm MMPNCC produces superior results mainly when the subsets present lower dimension. In addition to this greater accuracy, the MMPNCC also has a lower value for variance, which shows the greater stability of the classifier. We can also consider that our algorithm is invariant to the dimensionality reduction obtaining the maximal accuracy value using only the fifty first main components of the feature's vector as can be seen in Table 8.

Table 7. Testing results to dataset MUSIC with 250 musics

	Linear SVM		**Polynomial SVM**		**Gaussian SVM**		**MMP NCC**	
dimension	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
50	64,79	10.154	60,15	9,117349	48,48	12,88557	68,51	4.159
100	66,26	10.408	59,14	8,509503	58,20	11,91191	67,54	3.489
250	66,20	10.744	58,23	8,59202	62,52	12,1056	70,01	3.157

Table 8. Testing results to dataset MUSIC with 500 musics

	Linear SVM		**Polynomial SVM**		**Gaussian SVM**		**MMP NCC**	
dimension	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
50	50,81	9.558	62,62	5,525174	38,36	12,14543	69.26	2.454
100	64,00	7.869	62,52	5,867435	46,48	13,20109	67.82	2.720
250	64,46	8.699	61,63	6,470109	57,46	10,78102	67.21	3.122
500	64,63	8.855	61,55	6,459904	58,02	10,09458	68.94	2.357

After evaluating the performance of the classifiers for 250 and 500 instances, we will present in the next tables the results obtained for the MUSIC dataset containing 1000 songs and 5 genres and for the dataset GTZAN containing 1000 songs and 10 genres. As we have already stated, in both datasets the feature vector was constructed in two distinct scenarios, one with audio segments containing 15 seconds and the other with 30 seconds.

Table 9 presents the results obtained for the MUSIC dataset with 1000 songs and 15 seconds. With the increasing of the training set size we do not observe an improvement in the test accuracy. However, the results for length segments of 30 seconds, depicted in Table 10, show a slight improvement in accuracy

results. Also, in the two scenarios, the best results are achieved with the fifty first main components.

Table 9. Testing results to dataset MUSIC with 1000 musics and 15s

	Linear SVM		**Polynomial SVM**		**Gaussian SVM**		**MMP NCC**	
dimension	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
50	34,69	13,7856	62,19	5,1639	31,43	15,4200	66,65	1,4668
100	49,77	12,1024	61,31	5,1188	37,38	14,5680	66,06	1,5403
250	62,88	7,4630	60,65	5,6704	44,39	12,5331	66,58	1,5243
500	62,54	7,4377	60,60	5,7184	45,94	12,3197	66.49	1.4035

Table 10. Testing results to dataset MUSIC with 1000 musics and 30s

	Linear SVM		**Polynomial SVM**		**Gaussian SVM**		**MMP NCC**	
dimension	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
50	37,26	14,8389	62,96	5,0256	33,62	16,1976	68,74	1,3828
100	52,82	11,2468	62,64	5,7247	40,98	15,4112	67,93	1,8604
250	65,09	8,4239	62,61	6,2422	50,08	12,5440	67,75	1,5058
500	65,32	8,2729	62,43	6,4261	55,57	10,7210	68.01	2.1396
1000	65,09	8,3828	62,18	6,3724	55,40	11,3932	68,30	1,3704

Table 11 and 12 present the results obtained for the GTZAN dataset with 1000 instances and respectively with segments length with 15 and 30 seconds. These results present the performance of the algorithms for a problem with more classes, which makes it more difficult to predict and consequently impacts the performance of the classifier. The SVM remained underperforming the MMPNCC algorithm. It is important to highlight that the Linear SVM overperforming the Polynomial and Gaussian kernels emphasizing that the problem of learning music similarity has a better solution with a classifier based on a linear hypothesis . SVM presented a worse performance for the smaller dimensions, following the behavior of the previous results, but the parameterized algorithm remains invariant and again achieved the better results with the first fifty main components. Likewise, we can observe a slight improvement in the accuracy when the audio segments with 30 seconds is used concluding that this length and the use of the first fifty components of the feature vector is a reasonable choice for the feature vector representation.

Table 11. Testing results to dataset GTZAN with 1000 musics and 15s

	Linear SVM		**Polynomial SVM**		**Gaussian SVM**		**MMP NCC**	
dimension	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
50	34,89	19,2909	56,64	9,2045	17,67	22,4392	62,23	3,9809
100	51,48	11,2299	57,03	9,6904	29,23	19,1642	61,98	3,0265
250	57,01	11,2727	55,68	9,9172	39,93	17,5515	62,37	3,2566
500	56,92	11,3784	56,01	10,2765	41,64	16,1536	61,77	2,7300

Table 12. Testing results to dataset GTZAN with 1000 musics and 30s

	Linear SVM		**Polynomial SVM**		**Gaussian SVM**		**MMP NCC**	
dimension	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
50	40,25	17,4473	56,86	8,2375	24,34	22,2084	63,11	2,4515
100	58,43	10,8763	57,76	10,0666	36,62	17,9047	63,46	3,1490
250	58,10	11,6376	56,52	9,7833	52,15	13,1300	62,23	2,5189
500	58,29	11,9865	56,44	10,7436	56,67	12,1089	61,58	3.3075
1000	57,70	11,9619	56,27	11,2594	56,34	11,7100	60.85	2.7999

Finally, we report the classification performance of the parameterized algorithm displaying the confusion matrices of the two experiments. In Figure 2 we have the best performance of the MMPNCC algorithm when applied to GTZAN dataset with 1000 musics and 30s. In Figure 3 we have the same best result when applied to MUSIC dataset with 1000 musics and 30s.

We can observe that the accuracy values and the error distribution along the different genres are very closer except for the classical genre that presents a well formed musical structure, ensuring the invariant algorithm property when applied to different datasets. However, the medium measure of performance, about 63,5% for 10 genres and 69% for 5 genres, pointed out that the music classification problem is a fuzzy problem where the genre clusters do not have distinct boundaries making difficult a better classification.

This can be demonstrated by looking the results of both confusion matrices wherein the largest misclassifications values occur when the genres have a more similar musical structure. For example, in GTZAN dataset we can highligth the misclassification between genres rock and metal. This is compatible with the fact that metal can be considered a sub-genre of rock. On the other hand, in MUSIC dataset, the misclassification between genres samba and jazz can be considered by the fact that these genres contain several common music elements.

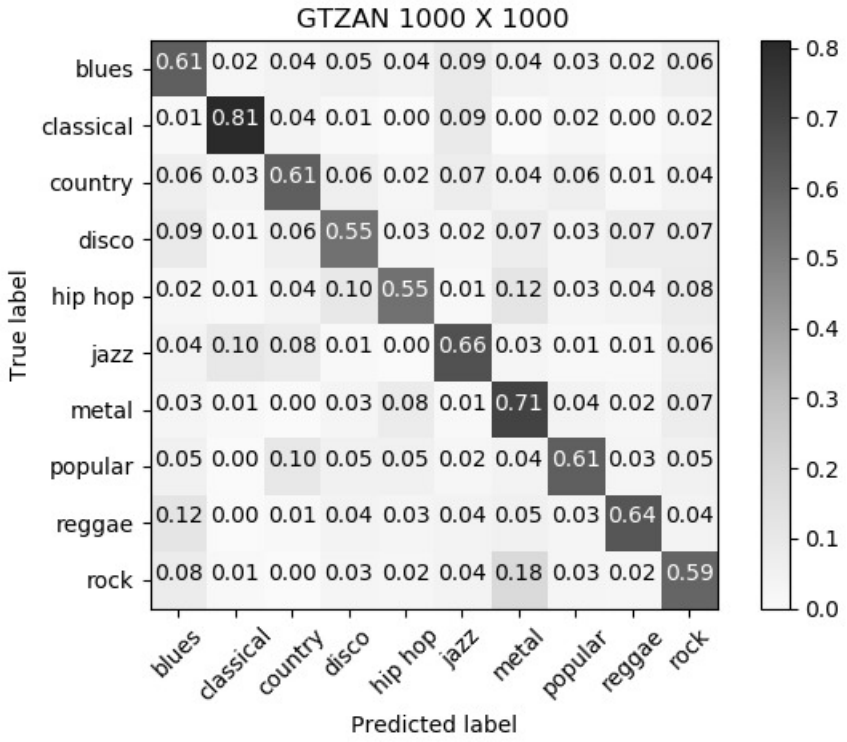

Figure 2: Confusion Matrix to GTZAN dataset with 1000 musics and 1000 dimensions

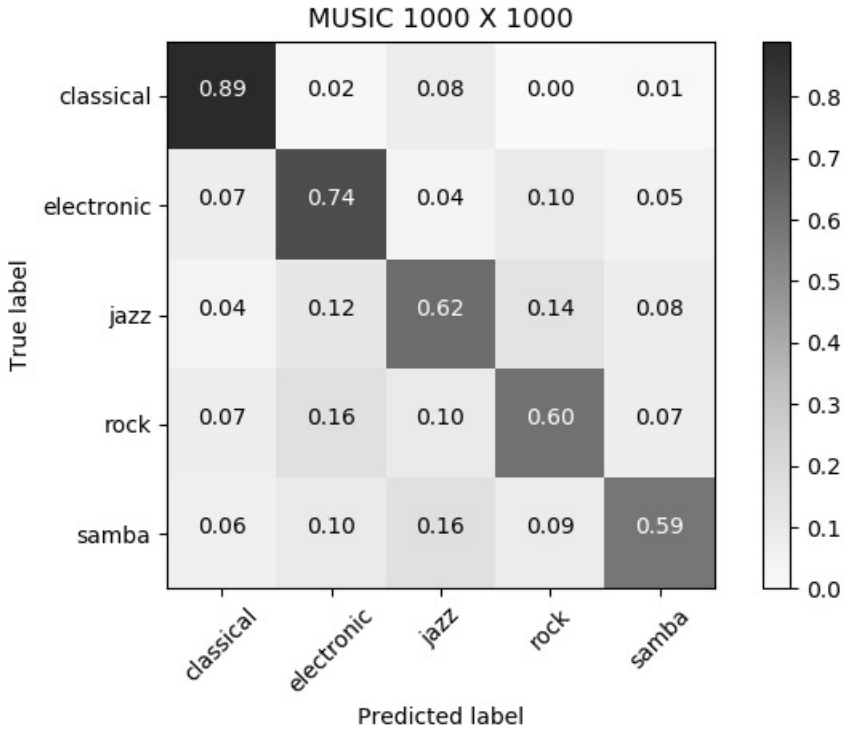

Figure 3: Confusion Matrix to MUSIC dataset with 1000 musics and 1000 dimensions

5.3. Comparative Analysis

Although the development of music similarity learning are carried in different settings with distinct approaches we present here a comparative analysis of our results with some works that also use the GTZAN dataset. In the work (Tzanetakis and Cook, (2002) the authors reported a study of feature analysis in a music classification process using GTZAN dataset. They used as features the timbral texture, rhythmic content and pitch content. The classification results are calculated using a ten-fold cross-validation technique where the dataset to be evaluated is randomly partitioned so that 10% is used for testing and 90% is used for training. Using the proposed feature sets, the authors obtain 61% of accuracy for ten musical genres.

In (Li et al., (2003) a comparison of the performance of several classifiers about a music classification scenario using the GTZAN dataset with various feature subsets is done. The accuracy values are also calculated via ten-fold cross-validation technique. The results obtained using MFCC features were: SVM one-against-one: 58.40%, SVM one-against-all: 58.10%, Gaussian Mixture Models (GMM):46.40%, Linear Discriminant Analysis (LDA): 55.50% and K-Nearest Neighbor (KNN): 53.70%.

Currently many authors consider the use of Deep Learning as the state of-the-art in several areas of machine learning, as for example in image and speech recognition. However, we can observe in the work of (Vishnupriya and Meenakshi, (2018) that the best results of test accuracy achieved to GTZAN dataset using MFCC as feature representation is around 47% with 80% of the data for training and only 20% for testing.

6. Conclusions and Future Work

In this work we addressed music similarity learning and propose a novel music classification model using structural acoustic information extracted directly from MP3 audio files.

Experiments show that the classification model based on metric learning tends to improve its overall training and testing performance, reaching predictions values consistent with the state of the art, overperforming the soft margin linear SVM. The higher variance presented by SVM indicates a large variation in the prediction of future data compromising directly the reliability of the related model.

Experiments have shown that reducing dimensionality in a metric learning scenario does not result in a large reduction in test accuracy and this allows us to work with a reduced feature vector making viable the music classification process in an online settings. The proposed model presented a stable performance with lower variance even with the increase of the number of musics and with the reduction of the dimensionality.

The results obtained with the GTZAN dataset are consistent with the results found in the literature, being superior in most of the referenced works and with a proposal that uses a reduced set of training. The performance of the metric learning model using 50% of the data for training and 50% for testing indicates that, with the increase in the number of training constraints, the model tends to better evolve its generalization power when compared to SVM and other referenced classifiers.

As future work, it is intended to develop an optimization model that captures the individual preferences of each user using an online setting. Although in this work it was possible to construct a quadratic decision function learning different parameters vectors for each class we refuse this alternative observing the fact that the Polynomial and Gaussian kernel underperforming the Linear kernel in SVM results. Also we think that the music learning similarity is influenced and induced by the user's preferences and not by the type of genre mainly if we consider a real time scenario application.

References

Barrington, L., Oda, R., Lanckriet, G., (2009). Smarter than Genius? Human Evaluation of Music Recommender Systems. In: International Society for Music Information Retrieval Conference, {ISMIR}. Vol. 9.

Kobe, Japan, pp. 357–362.
URL http://ismir2009.ismir.net/proceedings/OS4-4.pdf

Bergstra, J., Casagrande, N., Erhan, D., Eck, D., Kégl, B., (2006). Aggregate Features and ADABOOST for Music Classification. Machine Learning 65 (2-3), 473–484.
URL http://dx.doi.org/10.1007/s10994-006-9019-7

Burred, J. J., Lerch, A., (2004). Hierarchical Automatic Audio Signal Classification. Journal of the Audio Engineering Society (JAES) 52, 724–739.
URL http://www.aes.org/e-lib/browse.cfm?elib=13015

Carafini, A., (2015). Quantização vetorial de imagens coloridas através do algortimo LBG. Ph.D. thesis, Federal University Rio Grande de Sul, Rio Grande do Sul, Brazil.

Coelho, M. A., Borges, C. C., Neto, R. F., (2017). Uso de predição estruturada para o aprendizado de métrica. In: Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods.

Coelho, M. A., Neto, R. F., Borges, C. C., (2012). Perceptron Models for Online Structured Prediction. In: Proceedings of the 13th international conference on Intelligent Data Engineering and Automated Learning. Vol. 7435. Springer-Verlag, Berlin, Heidelberg, pp. 320–327.
URL http://dx.doi.org/10.1007/978-3-642-32639-4{_}39

Cortes, C., Vapnik, V., (1995). Support-Vector Networks. Machine Learning 20 (3), 273–297.
URL https://doi.org/10.1007/BF00994018

Duda, R. O., Hart, P. E., (1973). Pattern Classification and Scene Analysis. John Willey & Sons, New York.

Edwards, A. W. F., Cavalli-Sforza, L. L., (1965). A Method for Cluster Analysis. Biometrics 21, 362–375.

Fisher, R. A., (1936). THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS. Annals of Eugenics 7 (2), 179–188.
URL http://dx.doi.org/10.1111/j.1469-1809.1936.tb02137.x

Giannakopoulos, T., Pikrakis, A., (2014). Audio Features. In: Introduction to Audio Analysis. Academic Press, Oxford, pp. 59–103.
URL http://www.sciencedirect.com/science/article/pii/B9780080993881000042

Gupta, S., (2014). Music Data Analysis: {A} State-of-the-art Survey. CoRR abs/1411.5.

Hadsell, R., Chopra, S., LeCun, Y., (2006). Dimensionality Reduction by Learning an Invariant Mapping. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). Vol. 2. pp. 1735–1742.

Herrada, O. C., (2010). The Long Tail in Recommender Systems. In: Music Recommendation and Discovery. Springer Berlin Heidelberg, pp. 87–107.

Li, T., Ogihara, M., Li, Q., (2003). A Comparative Study on Content-based Music Genre Classification. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. SIGIR '03. ACM, New York, NY, USA, pp. 282–289.
URL http://doi.acm.org/10.1145/860435.860487

Linde, Y., Buzo, A., Gray, R., (1980). An Algorithm for Vector Quantizer Design. IEEE Transactions on Communications 28 (1), 84–95.

Lloyd, S., (1982). Least squares quantization in PCM. IEEE Transactions on Information Theory 28 (2), 129–137.

Loughran, R., Walker, J., O'Neill, M., O'Farrell, M., (2008). The Use of Mel-frequency Cepstral Coefficients in Musical Instrument Identification. In proceedings of the international computer music conference, 24–29.

McFee, B., Barrington, L., Lanckriet, G., (2010). Learning Similarity from Collaborative Filters. In: Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR. pp. 345–350.

McKinney, M. F., Breebaart, J., (2003). Features for audio and music classification. International Society for Music Information Retrieval Conference, ISMIR, 151–158.

Oppenheim, A. V., (1969). Speech Analysis-Synthesis System Based on Homomorphic Filtering. The Journal of the Acoustical Society of America 45, 458–465.
URL https://doi.org/10.1121/1.1911395

Pampalket, E., (2006). Computational models of music similarity and their application in music information retrieval. Ph.D. thesis, Vienna University of Technology, Vienna, Austria.
URL http://www.ofai.at/{~}elias.pampalk/publications/pampalk06thesis.pdf

Schapire, R. E., Singer, Y., (1999). Improved Boosting Algorithms Using Confidence-rated Predictions. Machine Learning 37 (3), 297–336.
URL https://doi.org/10.1023/A:1007614523901

Schultz, M., Joachims, T., (2004). Learning a Distance Metric from Relative Comparisons. In: Thrun, S., Saul, L. K., Schölkopf, B. (Eds.), Advances in Neural Information Processing Systems 16. MIT Press, pp. 41–48.
URL http://papers.nips.cc/paper/2366-learning-a-distance-metric-from-relative-comparisons.pdf

Siqueira, J. K., (2012). Reconhecimento de voz contínua com atributos mfcc, ssch e pncc, wavelet denoising e redes neurais. Ph.D. thesis, PUC RIO DE JANEIRO, Rio de Janeiro, Brazil.

Slaney, M., Weinberger, K., White, W., (2008). Learning a metric for music similarity. In: International Conference on Music Information Retrieval. pp. 313–318.

Southard, D. A., (1992). Compression of digitized map images. Computers & Geosciences 18 (9), 1213–1253.
URL http://www.sciencedirect.com/science/article/pii/009830049290041O

Stevens, S. S., Volkmann, J., Newman, E. B., (1937). A scale for the measurement of the psychological magnitude pitch. Journal of the Acoustical Society of America 8, 185–190.

Sturm, B. L., (2013). The {GTZAN} dataset: Its contents, its faults, their effects on evaluation, and its future use. CoRR abs/1306.1.

Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C., (2005). Learning Structured Prediction Models: A Large Margin Approach. In: Proceedings of the 22Nd International Conference on Machine Learning. ICML '05. ACM, New York, NY, USA, pp. 896–903.
URL http://doi.acm.org/10.1145/1102351.1102464

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

Villela, S. M., de Castro Leite, S., Neto, R. F., (2016). Incremental p-margin algorithm for classification with arbitrary norm. Pattern Recognition 55, 261–272.
URL http://www.sciencedirect.com/science/article/pii/S0031320316000376

Vishnupriya, S., Meenakshi, K., (2018). Automatic Music Genre Classification using Convolution Neural Network. In: 2018 International Conference on Computer Communication and Informatics (ICCCI). pp. 1–4.

Vlegels, J., Lievens, J., (2017). Music classification, genres, and taste patterns: A ground-up network analysis on the clustering of artist preferences. Poetics 60, 76–89.
URL http://www.sciencedirect.com/science/article/pii/S0304422X16301930

Wolff, D., Weyde, T., (2014). Learning music similarity from relative user ratings. Information Retrieval 17 (2), 109–136.
URL https://doi.org/10.1007/s10791-013-9229-0

Xin, L., Xuezheng, L., Ran, T., Youqun, S., (2014). Content-based retrieval of music using mel frequency cepstral coefficient (MFCC). Computer Modelling & New Technologies 18 (11), 1356–1361.
URL http://www.cmnt.lv/upload-files/ns{_}79art225.pdf

Xing, E. P., Ng, A. Y., Jordan, M. I., Russell, S., (2002). Distance Metric Learning, with Application to Clustering with Side-information. In: Proceedings of the 15th International Conference on Neural Information Processing Systems. Vol. 15 of NIPS'02. MIT Press, Cambridge, MA, USA, pp. 521–528.
URL http://dl.acm.org/citation.cfm?id=2968618.2968683

Yen, F. Z., Luo, Y.-J., Chi, T.-S., (2014). Singing Voice Separation Using Spectro-Temporal Modulation Features. In: Proceedings of the 15th International Society for Music Information Retrieval Conference, {ISMIR}. pp. 617–622.