Expert Systems With Applications 144 (2020) 113071

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A Music Classification model based on metric learning applied to MP3 audio files

Angelo Cesar Mendes da Silva[a], Maurício Archanjo Nunes Coelho[b], Raul Fonseca Neto[a,*]

[a] Departament of Computer Science, Universidade Federal de Juiz de Fora, Brazil
[b] Academic Department of Computer Science, IF Sudeste MG - Rio Pomba, Brazil

ARTICLE INFO

Article history:
Received 2 February 2019
Revised 31 October 2019
Accepted 1 November 2019
Available online 4 November 2019

Keywords:
Music similarity
Metric learning
Feature extraction
Mel frequency cepstral coefficient
Principal components analysis

ABSTRACT

The development of models for learning music similarity 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 whose main objective is to learn a personalized metric for each customer. 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 according to the users taste. The structured prediction solution aims to maximize the separation margin between genre centroids and to minimize the overall intra-cluster distances. To extract the acoustic information we use the Mel-Frequency Cepstral Coecient (MFCC) and made a dimensionality reduction using Principal Components Analysis (PCA). 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). Also, to prove the prediction capacity, we compare our results with two recent works with good prediction results on the GTZAN dataset. Experiments show promising results and encourage the future development of an online version of the learning model to be applied in streaming platforms.

Published by Elsevier Ltd.

1. Introduction

Knowledge about customers preference or users profile allows the opportunity to offer products in a personalized way and, consequently, improve the probability of a customer to acquire a particular product or service. In the millionaire market of music streaming platforms, learning the customers preference is crucial for reducing the churn rate of clients and the cost of acquisition of new subscribers. Among the tactics to retain the customer, these platforms frequently use techniques as recommendation systems offering single options or playlists. In general, these recommendations are based on expert-added tags (Barrington, Oda, & Lanckriet, 2009) or on collaborative filters (McFee, Barrington, & Lanckriet, 2010).

Works based on expert-added tags are sensible to the subjective of tags attributed to the music and collaborative filters are underperformed if the imbalance between the number of users and the evaluations contained in the database is large(Barrington et al., 2009; McFee et al., 2010). In this sense, we propose a novel approach for learning the customers preference estimating music similarity based on metric learning where all information is extracted directly from the MP3 audio files. Specifically, we consider for each music sample the use of a thirty seconds long audio segment and extract a feature vector from the audio segment using the Mel-Frequency Cepstral Coeficient (Loughran, Walker, O'Neill, & O'Farrell, 2008). Due to the large number of extracted features we made a study of dimensionality reduction using the Principal Components Analysis instead of a Feature Selection approach. The experiments, varying the number of attributes, show that the learning algorithm is almost invariant with respect to the number of attributes.

In addition to MFCC several works made a study of other types of temporal acoustic features. For example, (Wolff & Weyde, 2014) used a set of low level (chroma and timbre vectors) and high level (loudness, beat and tatum means and variances) features, (McKinney & Breebaart, 2003) made a comparative study involving four groups of audio features (low-level signal, MFCC, psychoacoustic and auditory model) and (Bergstra, Casagrande, Erhan, Eck, & Kégl, 2006) included a set of audio features from different methods of audio signal processing such as MFCC, Fast Fourier

Abbreviations: RCEPS, Real Cepstral Coefficients; ZCR, Zero-crossing Rate; AFTE, Auditory Filterbank Temporal Envelopes; CHR, Chromagram; LSP, Line Spectral Pairs; TMBR, Timbre; SCF, Spectral Crest Factor; SFM, Spectral Flatness Measure.

* Corresponding author.
E-mail address: raulfonseca.neto@ufjf.edu.br (R.F. Neto).

https://doi.org/10.1016/j.eswa.2019.113071
0957-4174/Published by Elsevier Ltd.

Transform (FFT), Real Cepstral Coefficients (RCEPS) and Zero-crossing Rate (ZCR). Certainly, as can be demonstrated in the cited works, the aggregation of other temporal features to MFCC can improve classifier accuracy. However, considering that the aim of our work is to learn music similarity based on metric learning, we did not make a detailed study of feature selection and opted to use only MFCC features to generate the audio file inputs.

The Metric Learning problem has been solved as an optimization problem and considers the minimization of a quadratic set of parameterized distances measured over pairs of samples and subject to triangular inequality constraints (Xing, Ng, Jordan, & Russell, 2002). Also, the distance values must be symmetrical and non-negatives. In this context, different solutions can be verified, such as learning a full parameter matrix or a diagonal matrix, resulting in a parameter vector. In the latter form 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, Chopra, & LeCun, 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, Borges, & Neto, 2017). In the context of genre classification, it is based on the fulfillment of a set of constraints that attests the pertinence of each music sample in relation to their respective genre centroid when compared to other alternatives. These constraints represent the inequality 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 & Weyde, 2014) also uses an analogous approach for learning the music similarity but the authors consider the learning of a distance metric from relative comparisons (Schultz & Joachims, 2004) involving for each constraint a triple of audio samples and therefore a cubic number of constraints. Certainly, the major theoretical contribution of our work is the reformulation of the Metric Learning problem based on pairwise relations (Xing et al., 2002) employing an equivalence theorem proved in (Edwards & Cavalli-Sforza, 1965). This theorem shows that the intra-cluster distance is proportional to sum of the distances between all pairs of samples that belong to the same cluster. In this sense, our model is able to model the Metric Learning problem using only a linear number of constraints.

We perform an extensive evaluation of the model 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 the one-against-all strategy. We also made experiments with variations on audio segment length, feature dimensionality and in the training set size in order to understand the robustness of the proposed model with respect to these parameters. Our experiments and results show that the metric learning from comparisons to genre centroids has a positive effect on the process of learning music similarity. In the experiments, we use two types of datasets. The first is the public GTZAN dataset that consists of 1000 audio segments with 30 s length each equally partitioned in 10 genres. The second dataset, named MUSIC, is an artificial dataset containing 1000 audio segments with 30 s each but distributed on only 5 genres. Also, to prove the prediction capacity, we compare our results with two recent works with good prediction results on the GTZAN dataset. The first employs a deep network for feature extraction coupled with a Random Forest Classifier (Sigtia & Dixon, 2014) and the second employ a Gaussian Process approach (Markov & Matsui, 2014).

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, audio contents 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 characteristics information about music instances. In this sense, it is possible to employ Machine Learning algorithms to associate feature vectors of instances with their classes for solving music classification tasks (Gupta, 2014).

The classification models based on audio content exploit the acoustic temporal features of the music presents in digital audio signals. In the symbolic level, the characteristics are extracted from symbolic data and presented at a higher level of abstraction. Those based on the lyrics use text mining techniques to extract information and execute a semantic analysis to make the classification. The use of metadata makes up solutions similar to those reported in the Collaborative Filters (CF) models, a technique commonly used to evaluates the music similarity exploring the users feedback information (Gupta, 2014; McFee et al., 2010).

In (Vlegels & Lievens, 2017) is reported the attempt to identify clusters of people that have similar relationship to the same favorite set of artists, singers and composers instead of specific music genres. The model was built on existing knowledge in social network analysis using users profile information from different socio-demographic characteristics and cultural behavior. This information is obtained from respondents and involves questions in a broad range of domains like arts, everyday culture, leisure activities, sport, and recreation. In this sense, a two-mode bipartite network was constructed with respondents in the rows entry and their favorites artists, singers and composers in the columns entry. For network analysis the model employs the Integrated Region Matching (IRM) technique to evaluate the overall similarity between clusters. The results show that models using information based only on cultural analysis and genre preferences might be inadequate or insufficient for a better classification because they miss important informations that cannot be captured. Also, the authors identified that the artists clusters do not follow predefined music genres boundaries.

Again, 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, the correct evaluation of music similarity plays a central role in music recovery, recommendation and classification tasks. Observe that, if we have an appropriated learned metric, we can return several options of music with similar characteristics, indicating new preferences and also label unknown samples (Pampalket, 2006; Slaney, Weinberger, & White, 2008; Wolff & Weyde, 2014).

In (Wolff & Weyde, 2014) the authors show that the similarity measure between a music sample and others is highly dependent on the context in which it 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 customers preference. Several approaches that use music similarity have a

common characteristic in which the users 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 play musics of one genre or musics 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). In this sense, these approaches cannot work well with a learning model based on users feedback.

Collaborative Filter aims to individualize the users profile based on the evaluations that they execute in the system. However, this technique presents several problems (Herrada, 2010) such as sparseness due to lack of sufficient evaluations in the base, subjectivity since users can differ in evaluations on the same data, and scalability because the complexity tends to increase proportionally in relation to the number of evaluations. Thereby, the major difficulty in evaluating music similarity based on information coming from CF 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. To overcome these problems, we have to use the audio content information with the intention of removing that subjectivity. This way, the feature vector extracted from audio information 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 from users preference without limiting itself to metadata used for music description (Correa & Rodrigues, 2016). It is expected that audio content allows us to learn the preference of the user in a more objective way since the information is collected in the form of music structural composition and its temporal 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 instances and represent the inequality condition that the parameterized distance between an instance (music sample) 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 instances are embedded together with these respective genre clusters. As previously mentioned, our work is similar to the model of relative comparisons proposed in (Wolff & Weyde, 2014) that have a Structured SVM approach (Schultz & 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. However, the major drawback of this formulation is the number of constraints that scales in order $O(n^3)$ with the number of instances.

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

Finally, we report two previously mentioned works used to compare with our results. The first employs a deep network for feature extraction coupled with a Random Forest Classifier (Sigtia & Dixon, 2014). The aim of the work is to improve the training time and overcome the problem of get stuck in local minima making the learning algorithm for deep network competitive and at the same time producing good results in terms of accuracy. The experiments were applied on the GTZAN and ISMIR 2004 datasets using a 4-fold cross-validation test. The second employs a Gaussian Process (GP) approach (Markov & Matsui, 2014) and compares the obtained results with the state-of-the-art SVM. The authors built two models, one for music genre classification and another for music emotion estimation. The music classification model also uses in the experiments the GTZAN dataset using a 10-fold cross-validation test. The obtained results clearly showed that the GP outperforms the SVM both in genre classification and in emotion estimation tasks.

3. Feature extraction

3.1. Mel frequency cepstral coefficient

The work of (McKinney & Breebaart, 2003) carried out a study on the impact that temporal and static behaviors of a set of features can have on the classification performance of general audios and genres of music. Among the features sets analyzed, two presented higher performance in classification tasks: Auditory Filterbank Temporal Envelopes (AFTE) and Mel Frequency Cepstral Coefficient (MFCC). Also, the works of (Bergstra et al., 2006), (Burred & Lerch, 2004) and (Yen, Luo, & Chi, 2014) highlight the use of MFCC features to construct the feature vector in music classification tasks.

According to (McKinney & 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: firstly, it is passed through a bank of 18 4th-order bandpass filters spaced logarithmically from 26 to 9795 Hz; then 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.

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

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

In addition, (McKinney & Breebaart, 2003) also analyzed other low level sets of features and Psychoacoustics characteristics and the MFCC set presented better results to classify both general audios and music genres with less complexity in the extraction process. However, when the AFTE set is used there is an improve-

Table 1
AFTE Features.

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

Table 2
MFCC Features.

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

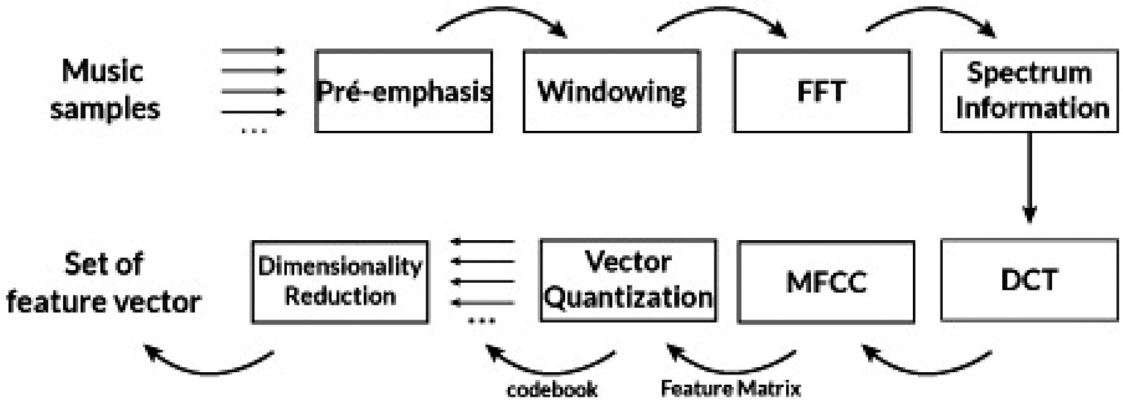

Fig. 1. Feature vector building process.

ment in the classification results, although it is not considered to be statistically significant. Due for this results, we opted to use only the MFCC technique for feature extraction. 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 and many other tasks (Pampalket, 2006).

The MFCC 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, Volkmann, & Newman, 1937). It aims to mimic the unique characteristics perceptible by the human ear. These coefficients are 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).

The number of coefficients to be used in the MFCC is another important issue in the extraction process due to the specific signal information that each represents. It is worth noting that, depending on the task at hand, different MFCCs subsets are adopted. For example, it has become usual in many music processing applications to select the first 13 MFCCs because they are considered sufficient to capture the discriminative information in the context of classification tasks(Giannakopoulos & Pikrakis, 2014).

According to (Pampalket, 2006), the extraction of 30 s of audio is enough to represent the information necessary to identify a music sample, and they should be extracted from the first half of the audio. We shifted 15 sto bypass a possible introduction, which may contain no information. The MFCC extraction process was done automatically using Librosa (McFee & Nieto, 2015)

3.2. Vector quantization

Vector quantization is a method usually applied in data compression. However, it also finds applications in the field of signal 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. The feature extraction process produces an n-dimensional feature vector for every piece of music. In our work we have considered only the first 13 MFCCs, where vector quantization is used to minimize the data of the extracted features. The vector quantization process was applied to the matrix of features, generating a codeword that represents each music sample. From here, every time we refer to the feature vector we are referring now to the quantized vector. Formally, 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 with N vectors, where $L \gg N$, contained in the same space R^k (Carafini, 2015). Therefore, we have:

$$Q : I \rightarrow C$$

where, $I = \{x_0, x_1, \ldots, x_{L-1}\}$ and $x_l \in R^k$, $C = \{y_0, y_1, \ldots, y_{N-1}\}$ and $y_i \in R^k$. 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 (Linde, Buzo, & Gray, 1980), also known as Generalized-Lloyd's Algorithm (GLA)(Southard, 1992).

3.3. Dimensionality reduction

Extracting MFCC features from audio segments makes the data volume extremely large, and different instances length can cause a variation in the dimensionality space. In the extraction process of MFCC features is used Discret Cossine Fourier (DCT). This transformation can reduce the number of coefficients generated after applying the specified parameterization techniques (Siqueira, 2012). The reduction is made through a property of DCT, known as energy compression, concentrating the most significant values in the first positions of the vector, opening a high possibility to reduce the dimensionality of the feature vector and, consequently, increasing the computational efficiency of the tasks. 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 ortho-normally so that the data variance remains constant, but is concentrated in lower dimensions. The matrix of data being transformed consists of one vector of coefficients for each audio segment. Thus, there is now a matrix representing all the data. The correlation matrix of the data matrix is then calculated. The principal components of the data set can be recovered from the eigenvectors of this correlation matrix. Then, we made a variance study of the components and concluded that the first five components concentrate most of the variance of the whole set, about 80%. To measure the impact and effects of the dimensionality reduction on our classification model, we made experiments varying the number of components. The maximum dimensionality of each data set created after the PCA analysis is limited to the number of music samples. For 30 s of music, the feature vector obtained has a dimensionality of 1293, and for 15 s the size is 647.

Every stage of the feature vector construction process, carried out in eight steps, is illustrated in the flowchart shown in Fig. 1.

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 $\{x_i, i = 1, \ldots, n\} \subset R^d$. 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: $S_1, S_2, \ldots, S_k$ each associated with a cluster. Therefore:

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

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

Otherwise, if the points are dissimilar, we have for a dissimilarity set D:

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

Generally, the Metric Learning problem 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 simple 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.

Let a measure of parameterized distance between two points defined as a function of a symmetric matrix A_{dxd} positive semidefinite (PSD) and not null:

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

with the following properties:

$d_A(x_i, x_j) > 0,$

$d_A(x_i, x_i) = 0,$

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

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

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:

$$Min \sum_l \sum_{(x_i, x_j) \in S_l} \| x_i - x_j \|_A^2 \quad (2)$$

subject to $A \succeq 0$ (PSD)

If we consider the parameters matrix a diagonal matrix, we have to learn a not null vector of parameters $w = [w_1, w_2, \ldots, w_d]$, 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 problem, we have a set of parameterized distances as a function of a full matrix A. For the diagonal matrix the PSD condition is satisfied if all components of vector w are non negatives. So, the equation (2) for the diagonal matrix can be reformulated as:

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

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

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

$$= \sum_l \eta_l \sum_{x_i \in S_l} w_1 (x_{i1} - c_{l1})^2 + w_2 (x_{i2} - c_{l2})^2 + \ldots + w_d (x_{id} - c_{ld})^2,$$

subject to $w_i \geq 0 \quad (3)$

where η_l represents the cardinality of S_l.

The equivalence between problems (2) and (3) is supported by (Edwards & Cavalli-Sforza, 1965) taking into account that the intra-cluster distance is proportional to the sum of distances between all pairs of points that belong to the same cluster. The proof of this theorem is based on the fact that each cluster centroid can be computed as the mean of the pertinent cluster vectors, that is:

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

To solve 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 closest to the work (Schultz & Joachims, 2004) that proposes an extension of Support Vector Machine (Cortes & 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 and x_k are similar or dissimilar. In this sense, it is 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:

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

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:

$$Min \frac{1}{2} \| w \|^2 + C. \sum_{i,j,k} \xi_{i,j,k} \quad (5)$$

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

$$\xi, w \geq 0$$

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 $S = \{x(i), y(i), i = 1, \ldots, m\}$ 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 $(x, y) \in S = \{x(i), y(i), i = 1, \ldots, m\}$, and the output y being subject to some constraint function $g(x, y)$. This inference problem is an inverse ill-posed problem. Therefore, the goal is to estimate the vector w such that $H_w(x)$ approximately, maps any desired output y. Thereby:

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

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, Chatalbashev, Koller, & Guestrin, 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 as a system of linear inequalities with the use of a variant of the Structured Perceptron algorithm (Coelho, Neto, & Borges, 2012).

Now, for this new approach, the update rule to correct a mistake without the loss function can be described as:

$$\text{for each pair } (x(i), y(i), i = 1, \ldots, m) \text{ do}$$
$$\text{if } (w.f_i(y_i) > w.f_i(y^*)), \text{ then}$$
$$w \leftarrow w - \eta(f_i(y_i) - f_i(y^*)), \tag{10}$$

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

Making 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.

Considering the fact that the set of linear inequations can presents several feasible solutions it is plausible 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, as in the problems (5) and (9). To implement the margin maximization process, we proposed the following formulation:

$$Max\ \gamma \tag{11}$$
$$\text{subject to: } w.(f_i(y*) - f_i(y_i)) \geq \gamma . \parallel w \parallel, i = 1, \ldots, m$$

where γ is the margin parameter.

Now, the new update rule can be described as:

$$\text{for each pair } (x(i), y(i)), i = 1, \ldots, m,$$
$$\text{if } (w.f_i(y_i) > w.f_i(y*) - \gamma \parallel w \parallel), \text{ then}$$
$$w \leftarrow w\left(1 - \frac{\eta\gamma}{\parallel w \parallel}\right) - \eta(f_i(y_i) - f_i(y*) \tag{12}$$

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 mistake, using the stochastic gradient method, 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, $d_w(x_i, c_l) = (x_i - c_l)^T W (x_i - c_l)$, we have to solve the following margin maximization problem:

$$Min \frac{1}{2} \parallel w \parallel^2 + C. \sum\nolimits_i \xi_i \tag{13}$$
$$\text{subject to: } d_w(x_i, c_k) - d_w(x_i, c_l) \geq 1 - \xi_i, \forall i = 1, \ldots, n,$$
$$\xi, w \geq 0$$

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 reformulated as:

$$Max\ \gamma \tag{14}$$
$$\text{subject to: } d_w(x_i, c_k) - d_w(x_i, c_l) + \lambda\alpha_i \geq \gamma . \parallel w \parallel, \forall i = 1, \ldots, n,$$
$$\alpha, w \geq 0$$

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, de Castro Leite, & Neto, 2016). Thus, the new update rule follows:

$$\text{for each pair } (x_i, c_l) \text{ do}$$
$$\text{if } d_w(x_i, c_k) - d_w(x_i, c_l) + \lambda\alpha_i < \gamma . \parallel w \parallel \text{ then}$$
$$w \leftarrow w\left(1 - \frac{\eta\gamma}{\parallel w \parallel} - \eta(d_w(x_i, c_k) - d_w(x_i, c_l)\right),$$
$$\alpha \leftarrow \alpha\left(1 - \frac{\eta\gamma}{\parallel w \parallel}\right)$$
$$\alpha_i \leftarrow \alpha_i + \eta \tag{15}$$

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 each 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

In this subsection, we describe the parameterized algorithms applied in the training and testing phases of the music classification task. Instead of use the K-means algorithm with Euclidean distances in an unsupervised setting we employ the Structured Perceptron algorithm that aims to learn an expert oriented metric. This algorithm is a maximal margin sample-by-sample version of the K-means with side information that adjusts the metric in supervised setting according to a set of predefined centroids. In this sense, we call this algorithm Maximal Margin Parameterized K-means (MMP K-means). For the testing phase, we employ the Nearest Centroid Classifier with parameterized distances using the metric learned in the training phase. We call this algorithm Maximal Margin Parameterized Nearest Centroid Classifier (MMP NCC).

The K-means algorithm minimizes the intracluster distance 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 that the sum of the distances between all pairs of vectors of the same cluster shares a similarity relation with the intracluster distance. Thus, the only necessary change in the Euclidean K-means algorithm is in the determination of the centers where now the parameterized distances to 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 centroid of each class, classifying the same according to the winner. On the other hand, the maximal Margin Parameterized Nearest Centroid Classifier (MMP NCC) uses parameterized distances for this purpose. If we consider a two-class classification problem with equal parameterized matrices we have as classification hypothesis a linear decision function. However, if we choose to learn two different parameters matrices, we have, as in the general case of a Fisher Discriminant 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 distributions 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:

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

According to (Cortes & 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 problem (18) when the covariance matrices are equal.

Table 3
GTZAN dataset.

samples	seconds	dimensions	classes
1000	15	5 - 50 - 100 - 250 - 500	10
1000	30	5 - 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	5 - 50 - 100 - 250 - 500	5
1000	30	5 - 50 - 100 - 250 - 500- 1000	5

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 binary 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 MMP NCC classifier:

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

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

5. Experiments and results

5.1. Datasets

In our work, we used two different datasets, depicted in Tables 3 and 4. One already known by several researchers in the area of Machine Learning and the other one was constructed by the authors. The former, named GTZAN[1], makes possible to compare the accuracy of our results with important 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 other words, with the customers musical taste. The GTZAN dataset appears in at least 100 published works and is the most-used public dataset for music similarity study (Sturm, 2013). The dataset consists of 1000 audio segments or pieces of musics each with 30 length. It contains 10 genres (Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Popular, Reggae, and Rock), each represented by 100 samples.

For the study of dimensionality reduction, we perform in MUSIC and GTZAN datasets a variance study that make possible to generate feature vectors with different dimensions ordering by the main components. Fig. 2 presents the accumulate variance in function of the numbers of components for both datasets. As can be see, we choose vectors size with 5, 50, 100, 250 and 1000 components, because the first 5 main components concentrate most of the variance of the whole set, about 80%.

For the study of parameterization we divided the MUSIC dataset into three nested subsets with respectively 250, 500 and 1000 audio segments. All subsets have the music samples equally distributed in 5 genres (Rock, Classical, Jazz, Electronic and Samba). For datasets with 1000 instances, we also construct subsets with 15 s length. From the GTZAN dataset we generate six subsets containing 1000 pieces of music with 30 s and 5, 50, 100, 250,

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

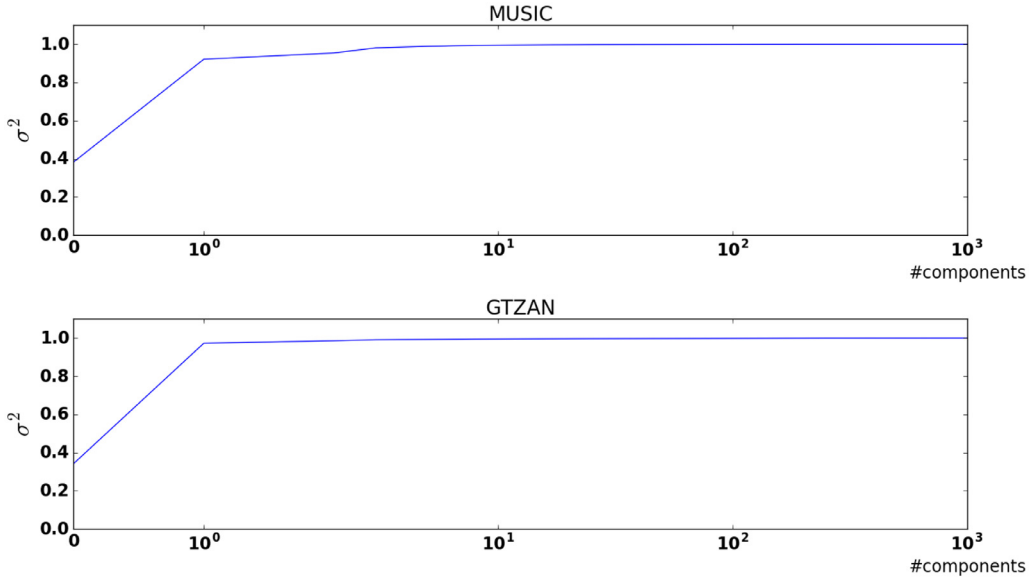

Fig. 2. Variance study using PCA.

Table 5
Training results in terms of accuracy(%) to MUSIC dataset with 250 music samples of 30s length.

dimension	Euclidean K-means		MMP K-means	
	μ	σ^2	μ	σ^2
50	32.52	5.461	55.00	6.514
100	37.14	5.002	56.14	5.437
250	34.30	4.907	60.10	3.264

Table 6
Training results in terms of accuracy(%) to MUSIC dataset with 500 music samples of 30s length.

dimension	Euclidean K-means		MMP K-means	
	μ	σ^2	μ	σ^2
50	3760	5.676	4120	6.902
100	3560	4.283	5160	5.311
250	3400	4.564	6200	2.880
500	3420	5.112	6360	4.935

500 and 1000 components and also five subsets with 1000 pieces of music with 15 s and 5, 50, 100, 250 and 500 components.

A summary of the two datasets with its variations is shown in Tables 3 and 4 reporting the number of samples, length, dimensions and number of classes.

5.2. Results

To evaluate the training performance we construct two scenarios, depicted in Tables 5 and 6, with MUSIC dataset using respective 250 and 500 samples. The results related to the training performance of the GTZAN dataset are not reported here. At this stage, we use this experiment to highlight the importance of the use of metric learning with side information in relation to the Euclidean metric. In this sense, we compare the results obtained by the parameterized algorithm (MMP K-means) with the Euclidean K-means. This analysis is fundamental to demonstrate the algorithms ability of learning music similarity according to the customers preference.

Notice that the classification error is considered when the difference of the distances in relation to the correct centroid has a negative value. The baseline algorithm Euclidean K-means 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.

To evaluate the testing performance we construct three scenarios. The first, with MUSIC dataset with 250 and 500 samples of 30s length, depicted in Tables 7 and 8. The second, also with MUSIC dataset however with 1000 samples of 15s and 30s length, depicted in Tables 9 and 10. The third, with GTZAN dataset with 1000 samples and also with 15s and 30s length, depicted in Tables 11 and 12. We compare the results obtained by the parameterized classifier algorithm (MMP NCC) against the state-of-the-art SVM algorithm with soft margin using Linear, Polynomial and Gaussian kernel functions. 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 we use an one-against-one strategy to construct the multi-classes 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 reported in Tables 7 to 12 represent the mean values for the accuracy and the variance obtained in each one of the experiments.

Tables 7 and 8 present the classification results obtained respectively by the dataset MUSIC with 250 and 500 samples compared with the SVM algorithm with the three types of kernel. Considering the variation on the dimensionality, the parameterized algorithm MMP NCC produces superior results against SVM, mainly when the subsets present lower dimension. In the sense, we can consider that our algorithm is invariant to dimensionality reduction, obtaining a good accuracy even when we use a smaller number of features, as can be seen in Table 8.

After evaluating the performance of the classifiers for 250 and 500 samples, we present in Tables 9 and 10 the results related to the MUSIC dataset containing 1000 samples and 5 genres. As well, we present in Tables 11 and 12 the results related to GTZAN dataset containing 1000 samples 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 s and the other with 30s length.

Table 7
Test results in terms of accuracy(%) to dataset MUSIC with 250 music samples of 30s length.

dimension	Linear SVM		Polynomial SVM		Gaussian SVM		MMP NCC	
	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
Test results in terms of accuracy(%) to dataset MUSIC with 500 music samples of 30s length.

dimension	Linear SVM		Polynomial SVM		Gaussian SVM		MMP NCC	
	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

Table 9
Test results in terms of accuracy(%) to dataset MUSIC with 1000 music samples of 15s length.

dimension	Linear SVM		Polynomial SVM		Gaussian SVM		MMP NCC	
	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
5	38.87	2.1179	36.86	2.5254	33.73	1.5922	64.86	1.7698
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
Test results in terms of accuracy(%) to dataset MUSIC with 1000 music samples of 30s length.

dimension	Linear SVM		Polynomial SVM		Gaussian SVM		MMP NCC	
	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
5	40.47	1.5750	38.61	2.7450	36.06	1.39393	68.16	2.2168
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
Test results in terms of accuracy(%) to dataset GTZAN with 1000 music samples of 15s length.

dimension	Linear SVM		Polynomial SVM		Gaussian SVM		MMP NCC	
	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
5	16.49	1.8350	17.53	1.8839	21.74	1.3659	62.29	3.4765
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
Test results in terms of accuracy(%) to dataset GTZAN with 1000 music samples of 30s length.

dimension	Linear SVM		Polynomial SVM		Gaussian SVM		MMP NCC	
	μ	σ^2	μ	σ^2	μ	σ^2	μ	σ^2
5	17.68	1.6929	17.66	2.3206	18.57	1.0847	61.04	4.1403
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

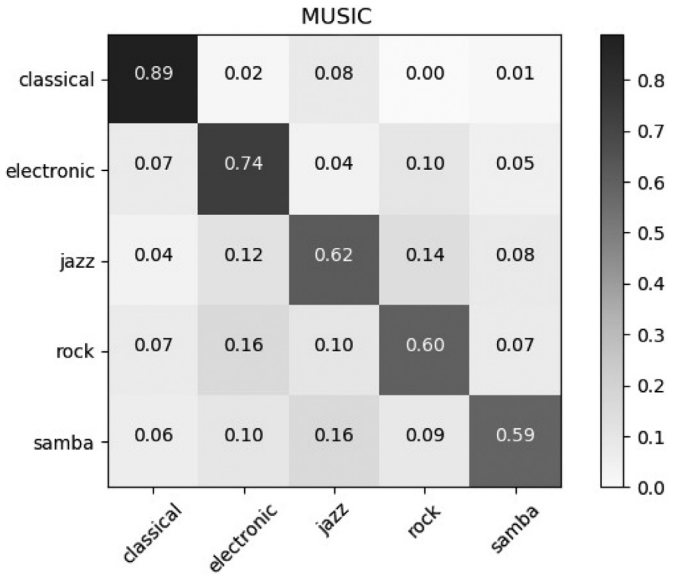

Fig. 3. Confusion Matrix to dataset MUSIC with 1000 music samples of 30s length and 5 features (50:50).

Analyzing the results reported in the Tables 9 to 12 we once again can conclude that the parameterized algorithm accuracy is invariant to dimensionality reduction and to audio segment length. However, the algorithm accuracy is sensible to the number of samples in the training size, showing a significant improvement in testing results when compared to Tables 7 and 8. It is important to highlight that, in all experiments, the SVM algorithm underperforms the MMP NCC algorithm. Also, the Linear SVM outperforms the Polynomial and Gaussian kernels, emphasizing that the problem of learning music similarity has a better solution with a classifier based on linear hypothesis.

Finally, we report the classification performance of the parameterized algorithm displaying the confusion matrices for three different scenarios. The first, depicted in Fig. 3, represent the best performance of the MMP NCC algorithm when applied to MUSIC dataset with 1000 musics of 30s length and 5 features. To evaluated the accuracy values we performed a 50:50 random validation test. 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. However, the mean accuracy value, about 69%, 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 exemplified by looking that the largest misclassifications values occur when the genres have a more similar musical structure. For example, in MUSIC dataset, the misclassification between genres samba and jazz can be considered higher by the fact that these genres contain several common music elements.

The second and third experiments, depicted in Figs. 4 and 5 respectively, are related to the GTZAN dataset with 1000 music samples of 30s length and 5 features considering, respectively, a 4-fold and a 10-fold cross-validation tests. These experiments will be commented in the next subsection.

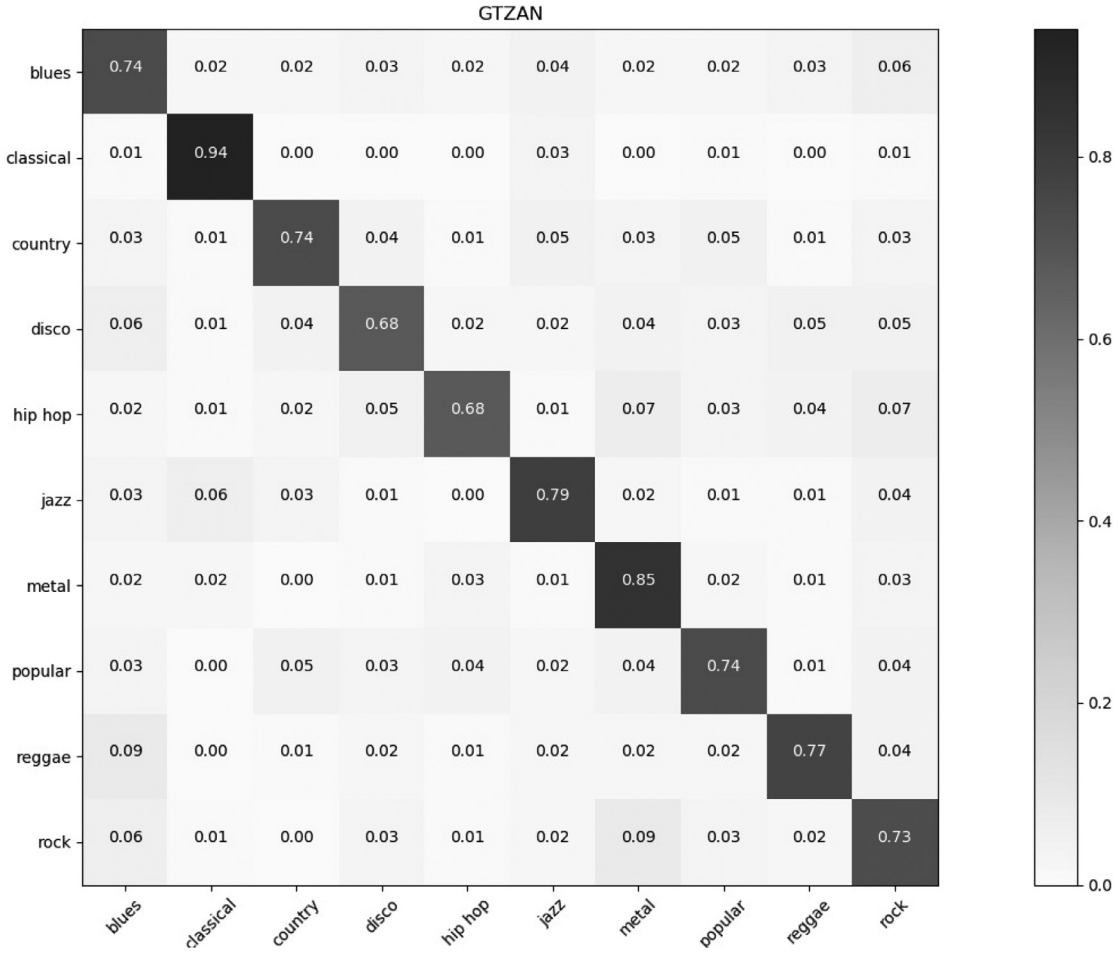

Fig. 4. Confusion Matrix to dataset GTZAN with 1000 music samples of 30s length and 5 features (4-fold).

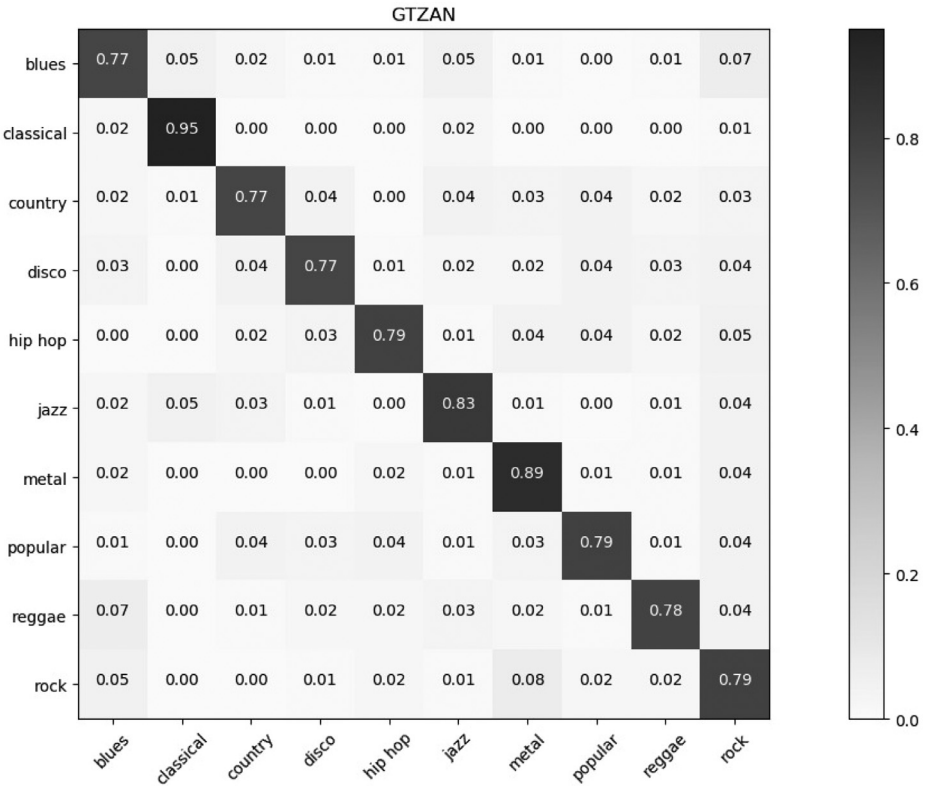

Fig. 5. Confusion Matrix to dataset GTZAN with 1000 music samples of 30s length and 5 features (10-fold).

5.3. Comparative analysis

The results obtained by MMP NCC algorithm with the GTZAN dataset are comparable to the results found in the literature, being superior in most of the referenced works and with a proposal that uses a reduced number of features. The results shown in Figs. 3 to 5 indicates that when we increment the number of training instances the model tends to improve its accuracy. Our accuracy results are: (61.04% ± 4.1403) to 50:50, (76.67% ± 6.9655) to 4-fold and (81, 34% ± 9.1726) to 10-fold cross-validation tests.

Although the study of music similarity is 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 & Cook, 2002) the authors reported a study of feature analysis in a music classification task. They used as features the timbral texture, rhythmic and pitch content. The classification results are calculated using a 10-fold cross-validation test. Using the proposed feature sets, the authors obtain 61% of accuracy for GTZAN dataset. In (Li, Ogihara, & Li, 2003) a comparison of the performance of several classifiers using the GTZAN dataset with various feature subsets is done. The accuracy values are also calculated via 10-fold cross-validation test. The results obtained using MFCC features were: SVM one-against-one: 58.40%, SVM one-against-all: 58.10%, Gaussian Mixture Models : 46.40%, Linear Discriminant Analysis: 55.50% and K-Nearest Neighbor: 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. The work presents in (Vishnupriya & Meenakshi, 2018) developed a Convolution Neural Network and the best accuracy result is achieved to Million Song Dataset using only MFCC features is 47% and using Mel Spec features is 76% with 80% of the data for training and only 20% for testing. The work presented in (Markov & Matsui, 2014) employs a Gaussian Process approach to make the music classification and uses a 10-fold cross-validation test. With only MFCC features the model obtain an accuracy value of 68.7%. Aggregating the CHR (chromagram), LSP (line spectral pairs), TMBR (timbre), SCF (spectral crest factor) and SFM (spectral flatness measure) features the accuracy value improves to 78.3%. However, this result is lower then the result related to our algorithm when uses a 10-fold cross-validation test, that is 81.34%.

Finally, we report the work presented in (Sigtia & Dixon, 2014) with employs a deep network for feature extraction coupled with a Random Forest classifier to perform the music classification

Table 13
Comparative analysis results in terms of accuracy(%) to GTZAN dataset.

classifier	length	features	type of features	validation test	accuracy(%)
Gaussian Process	30s	52 388	MFCC MFCC + Aggregation	10-fold	65.7 78.3
Deep Network + Random Forest	30s	513*	MFCC	4-fold	80.5
Metric Learning (MMP NCC)	30s	5	MFCC	4-fold 10-fold	76.7 81.3

* feature selection process.

task. The experiments carried out with only MFCC features using a 4-fold cross-validation test achieve 80.5% of accuracy using as input a 513 dimensional feature vector. This result is better than the result related to our algorithm that achieves 76.7% of accuracy keeping the same experimental setting. However, this superior result can be justified by the fact that the deep architecture was optimized in order to select from the raw data the better set of features. The Table 13 shows a summary of the comparative analysis related to the GTZAN dataset.

6. Conclusions and future work

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

The parameterization study demonstrated that, with a metric learning approach, the dimensionality reduction does not affect the test accuracy allowing us to work with a reduced feature vector. The proposed model also presented a stable performance even when we reduce the training set size and the audio segment length.

The results obtained with the GTZAN dataset are consistents with the results found in the literature, being superior in most of the referenced works. The performance of the metric learning model using 50% of the data for training and 50% for testing indicates that, with the increment in the number of training constraints, the model tends to better evolve its generalization power when compared to SVM and others referenced classifiers. This is justified by the accuracy results obtained by the model when the four and ten-fold cross-validation tests were used.

As future work, it is intended to develop an optimization model that captures the individual preferences of each user using an online setting to be applied in real time scenarios, for instance, in streaming platforms. Even though the proposed model performed well with genre classification, we believe that music learning similarity is influenced and induced directly by the user's preferences. In this sense, our method can be easily adapted to learn an individual metric in an online and personalized setting extend our application to other type of tasks, for example, playlist generation.

Declaration of Competing Interest

The author(s) declare(s) that there is no conflict of interest.

Acknowledgments

We would like to thanks Yuri Resende Fonseca for comments, suggestions and discussion in this paper that led to substantial improvements in the manuscript.

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}: 9* (pp. 357–362). Kobe, Japan

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. doi: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. 10.1.1.2.6582

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

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: 7435* (pp. 320–327). Berlin, Heidelberg: Springer-Verlag. doi:10.1007/978-3-642-32639-4_39.

Correa, D. C., & Rodrigues, F. A. (2016). A survey on symbolic data-based music genre classification. *Expert Systems with Applications, 60*, 190–210 https://doi.org/10.1016/j.eswa.2016.04.008.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273–297. doi:10.1007/BF00994018.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene analysis*. New York: John Willey & Sons.

Edwards, A. W. F., & Cavalli-Sforza, L. L. (1965). A method for cluster analysis. *Biometrics, 21*, 362–375. doi:10.2307/2528096.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*(2), 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x.

Giannakopoulos, T., & Pikrakis, A. (2014). Audio features. In *Introduction to audio analysis* (pp. 59–103). Oxford: Academic Press. https://doi.org/10.1016/B978-0-08-099388-1.00004-2.

Gupta, S. (2014). *Music data analysis: {A} state-of-the-art survey*. arXiv: 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): 2* (pp. 1735–1742). doi:10.1109/CVPR.2006.100.

Herrada, O. C. (2010). The long tail in recommender systems. In *Music recommendation and discovery* (pp. 87–107). Springer Berlin Heidelberg. doi:10.1007/978-3-642-13287-2.

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*. In *SIGIR' 03* (pp. 282–289). New York, NY, USA: ACM. doi: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. doi:10.1109/TCOM.1980.1094577.

Loughran, R., Walker, J., O'Neill, M., & O'Farrell, M. (2008). The use of mel-frequency cepstral coefficients in musical instrument identification. In *In Proceedings of the International Computer Music conference* (pp. 24–29).

Markov, K., & Matsui, T. (2014). Music genre and emotion recognition using gaussian processes. *IEEE Access, 2*, 688–697. doi:10.1109/ACCESS.2014.2333095.

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).

McFee, B., Raffel, C., Liang, D., Ellis, Daniel P. W., McVicar, M., Battenbergk, E., & Nieto, O. (2015). librosa: Audio and music signal analysis in python. In *In Proceedings of the 14th Python in Science Conference* (pp. 18–25). doi:10.5281/zenodo.591533.

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. doi:10.1121/1.1911395.

Pampalket, E. (2006). *Computational models of music similarity and their application in music information retrieval*. Ph.D. thesis. Vienna, Austria: Vienna University of Technology.

Schapire, R. E., & Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Machine Learning, 37*(3), 297–336. doi:10.1023/A:1007614523901.

Schultz, M., & Joachims, T. (2004). Learning a distance metric from relative comparisons. In S. Thrun, L. K. Saul, & B. Schölkopf (Eds.), *Advances in neural information processing systems 16* (pp. 41–48). MIT Press.

Sigtia, S., & Dixon, S. (2014). Improved music feature learning with deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 6959–6963). doi:10.1109/ICASSP.2014.6854949.

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

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 https://doi.org/10.1016/0098-3004(92)90041-O.

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. doi:10.1121/1.1915893.

Sturm, B. L. (2013). The {GTZAN} dataset: its contents, its faults, their effects on evaluation, and its future use. *CoRR, abs/1306.1*. doi:10.1080/09298215.2014.894533.

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*. In *ICML' 05* (pp. 896–903). New York, NY, USA: ACM. doi: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. doi:10.1109/TSA.2002.800560.

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 https://doi.org/10.1016/j.patcog.2016.01.016.

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). doi:10.1109/ICCCI.2018.8441340.

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 https://doi.org/10.1016/j.poetic.2016.08.004.

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

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*. In *NIPS'02: 15* (pp. 521–528). Cambridge, MA, USA: MIT Press.

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).