MUSICAL GENRE CLASSIFICATION OF AUDIO SIGNALS USING GEOMETRIC METHODS

Michal Genussov and Israel Cohen

Department of Electrical Engineering, Technion - Israel Institute of Technology
Technion City, Haifa 32000, Israel
email: {mgenus@tx, icohen@ee}.technion.ac.il

ABSTRACT

Musical genres are categorical labels characterizing pieces of music. Automatically classifying music into genres is gaining importance as a way to structure and organize the increasingly large numbers of music files available digitally on the web. In this work such a classification algorithm is developed and examined. The algorithm uses a vector of features based on the timbral texture of the music, and maps it into a new Euclidean space, by a non-linear method called "Diffusion Maps", before the classification stage itself. This method allows dimensionality reduction while preserving and emphasizing the distinction between different genres. The proposed classifier classifies accurately 97% when classifying 2 musical genres, and 52% when classifying 10 musical genres. This is compared to an accuracy of 88% and 28% respectively, when classifying without the proposed mapping.

1. INTRODUCTION

Musical genres are labels created and used for categorizing and describing the vast universe of music. Different genres differ from each other in their instrumentation, rhythmic structure and pitch content of the music. They include, for example - classic music, jazz, rock etc. In recent years there is a growing interest in automatically categorizing music into genres, as part of extracting musical information in general. Automatically extracting musical information is gaining importance as a way to structure and organize the increasingly large numbers of music files available digitally on the web. In addition, features evaluated by automatic genre classification can be used for tasks as similarity retrieval, classification, segmentation and audio thumbnailing.

In existing methods, the process of genre classification is composed of two steps: in the first step, relevant features (that represent the instrumentation, rhythmic structure, pitch content, etc.) are extracted from the signal, and a vector of features (feature-vector) is built. In the second step, a classification algorithm is applied on the feature-vector, such as k-nearest neighbors (k-nn) or Gaussian mixture model [6], support vector machines [7] or neural networks [4].

There are two fundamental problems in these methods:

1. In order to capture optimally the nature of the signal and differ efficiently between genres, the feature-vector usually needs to be high dimensional. As the number of signals increases, the computational complexity increases as well, leading to the need of a dimensionality reduction technique.
2. The traditional classification techniques, applied directly on the feature-vectors, might yield poor results if the feature-vectors lie in a non-linear manifold, in which Euclidean distances do not represent the intrinsic distances between them.

In this work we try to solve these problems using a technique called "Diffusion Maps" [3, 1]. We add an intermediate step to the process of classification - a step of dimensionality reduction of the feature space, before the classification operation itself. This technique performs a nonlinear reduction of the dimension by providing a parametrization of the data set on a lower-dimensional manifold, while emphasizing the differences between feature-vectors of different genres.

Another task which is dealt with, is the out-of-sample extension problem. A method called "Geometric Harmonics" [2] allows to reduce the computational complexity when building the classifier, by extending the parametrization of diffusion maps from a limited training data set to the rest of the data set. Furthermore, it embeds each new song we wish to classify, into the diffusion maps parametrization of the training set.

This paper is organized as follows: In Section 2 the classification algorithm is described, in Section 3 experimental results are presented and analyzed, followed by conclusions in Section 4.

2. THE CLASSIFICATION ALGORITHM

The classification algorithm is applied in three steps:

1. Feature extraction - a characteristic vector is defined for each song. It captures the essence of the timbre and texture (the "color" of the sound).
2. Dimensionality reduction - the data is embedded into a lower dimensional subspace. It is parameterized in a lower dimensional manifold using diffusion maps and geometric harmonics algorithms [3, 1, 2].
3. Classification - the data is classified according to its new parametrization using k-nearest neighbors algorithm.

Each of these steps is described in details:

2.1 Feature Extraction

The features used to characterize the songs are timbral texture features [6], which represent the spectral and temporal characters of the songs. They are defined over 30s time windows, called "texture windows", and include the mean and the variance of different coefficients calculated over short "analysis windows" of 15ms, during the 30s texture windows. The coefficients calculated over the analysis windows are:

1. *Spectral Centroid*: The spectral centroid is defined as the center of gravity of the magnitude spectrum of the STFT

 ISSN 2076-1465

$$C_t = \frac{\sum_{n=1}^{N}(M_t[n]\cdot n)}{\sum_{n=1}^{N}(M_t[n])}. \quad (1)$$

where $M_t[n]$ is the magnitude of the Fourier transform at frame t and frequency bin n.

2. *Spectral Rolloff*: The spectral rolloff is defined as the frequency R_t below which 85% of the magnitude distribution is concentrated

$$\sum_{n=1}^{R_t} M_t[n] = 0.85 \cdot \sum_{n=1}^{N} M_t[n]. \quad (2)$$

The rolloff is another measure of spectral shape.

3. *Spectral Flux*: The spectral flux is defined as the squared difference between the magnitudes of successive spectral distributions

$$F_t = \sum_{n=1}^{N}(M_t[n] - M_{t-1}[n])^2. \quad (3)$$

The spectral flux is a measure of the amount of local spectral change.

4. *Time Domain Zero Crossings*:

$$Z_t = \frac{1}{2}\sum_{n=2}^{N}|sign(x[n]) - sign(x[n-1])|. \quad (4)$$

where the *sign* function is 1 for positive arguments and 0 for negative arguments, and $x[n]$ is the time domain signal for frame t. Time domain zero crossings provides a measure of the noisiness of the signal.

5. *Mel-Frequency Cepstral Coefficients*: Mel-frequency cepstral coefficients (MFCC) are also based on the STFT. After taking the log-amplitude of the magnitude spectrum, the FFT bins are grouped and smoothed according to the Mel-frequency scaling. A discrete cosine transform is performed on the result, and the first five coefficients are taken.

Another feature is the *Low-Energy Feature*. It is defined as the percentage of analysis windows that have less RMS energy than the average RMS energy across the texture window. Tzanetakis and Cook [6] also extracted rhythmic and pitch content features, but they don't improve the results of our proposed classifier, and therefore are not used in this work.

To summarize, the feature-vector consists of the following 19 timbral texture features: low energy, means and variances of spectral centroid, rolloff, flux, zero-crossings over the texture window, and means and variances of the first five MFCC coefficients over the texture window.

2.2 Embedding the data into a lower dimensional space

Let (X, A, μ) be a measure space. The set X is the high-dimensional data set and the function μ represents the distribution of the points on X.

In addition to this structure, suppose that we are given a kernel function $k : X \times X \to R$ that satisfies, for $(x,y) \in X$:

- It is symmetric: $k(x,y) = k(y,x)$
- It is positive semi-definite: $k(x,y) \geq 0$

The kernel is a similarity function between two points of X, and it constitutes our prior definition of the *local* geometry of X. This is a major difference from *global* methods for dimensionality reduction, like principal component analysis, where all correlations between data points are taken into account [5]. The pair (X,k) define a graph in an Euclidean space. Following classical construction in spectral graph theory, a Markov random walk on the graph is defined:

$$p(x,y) = \frac{k(x,y)}{d(x)}. \quad (5)$$

where $d(x) = \int_X k(x,y)\, d\mu(y)$. The function p can be viewed as the transition kernel of a Markov chain on X, since $\int_X p(x,y)\, d\mu(y) = 1$. The operator P is defined by

$$Pf(x) = \int_X p(x,z) f(z) d\mu(z).$$

The expression $p(x,y)$ represents the probability of transition in one time step from node x to node y and it is proportional to $k(x,y)$. Accordingly, the probability of transition from node x to node y in t time steps is given by $p_t(x,y)$, which is the kernel of the t_{th} power P^t of P. Running the chain forward in time, or equivalently, taking larger powers of P, reveals geometric structures of X at larger scales. The random walk exhibits some important mathematical properties:

- The Markov chain has a stationary distribution given by

$$\pi(y) = \frac{d(y)}{\sum_{z\in X} d(z)}.$$

If the graph is connected, which we now assume, then the stationary distribution is unique [1].

- The chain is reversible:

$$\pi(x)p(x,y) = \pi(y)p(y,x).$$

- If X is finite and the graph of the data is connected, then the chain is ergodic [1].

If we apply spectral decomposition, it can be shown [1] that P has a discrete sequence of eigenvalues $\{\lambda_l\}_{l\geq 0}$ and eigenfunctions $\{\psi_l\}_{l\geq 0}$ such that $1 = \lambda_0 > |\lambda_1| \geq |\lambda_2| \geq ...$ and $P\psi_l = \lambda_l \psi_l$.

Now we relate the spectral properties of the Markov chain to the geometry of the data set X. In order to compute the powers of the operator P, we could use the eigenvectors and eigenvalues of P. Instead, we will directly employ these objects in order to characterize the geometry of the data set X. The family of *diffusion distances* $\{D_t\}_{t\in\mathbb{N}}$ is defined by

$$D_t(x,y)^2 \doteq \|p_t(x,\cdot) - p_t(y,\cdot)\|^2_{L^2(X,d\mu/\pi)} = \int_X (p_t(x,u) - p_t(y,u))^2 \frac{d\mu(u)}{\pi(u)}. \quad (6)$$

For a fixed value of t, D_t defines a distance on the set X, which reflects the connectivity in the graph of the data. The distance $D_t(x,y)$ will be small if there is a large probability

of transition from x to y. It is shown in [1] that $D_t(x,y)$ can be computed using the eigenvectors and eigenvalues of P:

$$D_t(x,y) = \left(\sum_{l \geq 1} \lambda_l^{2t} (\psi_l(x) - \psi_l(y))^2 \right)^{\frac{1}{2}}.$$

Since the eigenvalues $\lambda_1, \lambda_2, \ldots$ tend to 0 and have a modulus strictly less than 1, the above sum can be computed to a preset accuracy $\delta > 0$ with a finite number of terms: We define $s(\delta,t) = max\{l \in \mathbb{N} \text{ such that } |\lambda_l|^t > \delta |\lambda_1|^t\}$, then, up to the precision δ we have

$$D_t(x,y) = \left(\sum_{l=1}^{s(\delta,t)} \lambda_l^{2t} (\psi_l(x) - \psi_l(y))^2 \right)^{\frac{1}{2}}. \quad (7)$$

The family of *diffusion maps* $\{\Psi_t\}_{t \in \mathbb{N}}$ is defined by:

$$\Psi_t(x) \doteq \begin{bmatrix} \lambda_1^t \psi_1(x) \\ \lambda_2^t \psi_2(x) \\ \vdots \\ \lambda_{s(\delta,t)}^t \psi_{s(\delta,t)}(x) \end{bmatrix} \quad (8)$$

Each component of $\Psi_t(x)$ is termed a *diffusion coordinate*. The diffusion map $\Psi_t : X \rightarrow \mathbb{R}^{s(\delta,t)}$ embeds the data set into an Euclidean space of $s(\delta,t)$ dimensions, so that in this space, the Euclidean distance is equal to the diffusion distance (up to the relative accuracy δ), or equivalently $\|\Psi_t(x) - \Psi_t(y)\| = D_t(x,y)$

Therefore, if the dimensionality of the data can be reduced to $s(\delta,t)$, then $D_t(x,y)$, provided by the family of the diffusion maps, captures the distance between nodes x and y in the manifold of dimension $s(\delta,t)$. As t increases, the spectrum decay is faster, and less dimensions need to be used [$s(\delta,t)$ is smaller].

In this work, the set X represents the set of feature-vectors of different songs. The kernel $k(x,y)$ was chosen to be Gaussian:

$$k(x,y) = exp(-||(x-y)./\sigma||)^2 \quad (9)$$

where σ is a vector that consists of elements proportional to the standard deviations of each of the features. The division of $(x-y)$ by σ is element-wise, leading to a *multi scale* embedding, which means a different normalization for each component in the feature-vector. The matrix P is used without taking powers ($t = 1$), and $s(\delta,t)$=10 , which means that the top 10 diffusion coordinates (which correspond to the largest eigenvalues that do not equal 1) are taken, so the family of diffusion maps is reduced to:

$$\Psi(x) = \begin{bmatrix} \lambda_1 \psi_1(x) \\ \lambda_2 \psi_2(x) \\ \vdots \\ \lambda_{10} \psi_{10}(x) \end{bmatrix} \quad (10)$$

2.3 Out-of-sample extension

The parametrization described in the previous subsection is conducted over a limited data set, to maintain a limited computational complexity. In order to extend the family of diffusion maps to the rest of the data, we use a method called

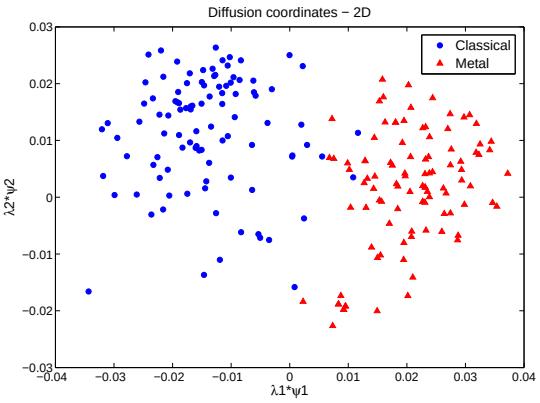

Figure 1: Diffusion coordinates (2D) of the classical and metal feature vectors

"geometric harmonics" ([3], [2]). If we denote the limited training data set, which was used to build the matrix P, as X, and the rest of the training data set as $\bar{X}$ ($X \subset \bar{X}$), then the extended eigenvectors which belong to the feature-vector $\bar{x} \in \bar{X}$ can be calculated as:

$$\bar{\psi}_j(\bar{x}) = \frac{1}{\lambda_j} \sum_{x \in X} p(x,\bar{x}) \psi_j(x) \quad (11)$$

and the new family of diffusion maps for each vector $\bar{x}$ is:

$$\bar{\Psi}(\bar{x}) = \begin{bmatrix} \lambda_1 \bar{\psi}_1(\bar{x}) \\ \lambda_2 \bar{\psi}_2(\bar{x}) \\ \vdots \\ \lambda_{10} \bar{\psi}_{10}(\bar{x}) \end{bmatrix} \quad (12)$$

Using this extension, the spectral analysis is performed only over a limited training data set, and then the diffusion coordinates of the rest of the training set are computed.

In order to classify new data from a set which will be denoted as $\tilde{X}$, the geometric harmonics method is applied again for every feature-vector $\tilde{x} \in \tilde{X}$, and the extended eigenvectors $\tilde{\psi}_j(\tilde{x})$ are calculated as in (11). The new family of diffusion maps $\tilde{\Psi}(\tilde{x})$ for each vector $\tilde{x}$ is given by

$$\tilde{\Psi}(\tilde{x}) = \begin{bmatrix} \lambda_1 \tilde{\psi}_1(\tilde{x}) \\ \lambda_2 \tilde{\psi}_2(\tilde{x}) \\ \vdots \\ \lambda_{10} \tilde{\psi}_{10}(\tilde{x}) \end{bmatrix}. \quad (13)$$

A new song is classified using the k-nearest neighbors (k-nn) method (k=5), where the corresponding family of diffusion maps $\{\tilde{\Psi}(\tilde{x})\}$ is classified according to the closest k nearest neighbors from the family of diffusion maps of the training set $\{\bar{\Psi}(\bar{x})\}$, and the measure distance for the k-nn is the Euclidean distance. We use k-nearest neighbors as the classifier because of its simplicity, and show that using the pre-stage of diffusion map yields good classification results even with such a simple classifier.

For visualization, the embedding of the feature vectors of the classical and metal genres to a 2D mapping is shown in Figure 1.

Table 1: Averaged Confusion Matrix Using Diffusion Coordinates - 10 Genres

	"Blues"	"Classic"	"Country"	"Disco"	"Hiphop"	"Jazz"	"Metal"	"Pop"	"Reggae"	"Rock"
Blues	**0.60**	0.00	0.09	0.04	0.05	0.02	0.07	0.00	0.04	0.10
Classic	0.01	**0.81**	0.05	0.01	0.00	0.08	0.01	0.00	0.00	0.03
Country	0.07	0.02	**0.48**	0.05	0.01	0.11	0.00	0.06	0.07	0.13
Disco	0.02	0.00	0.08	**0.37**	0.11	0.03	0.07	0.12	0.06	0.14
Hiphop	0.04	0.00	0.02	0.08	**0.52**	0.00	0.02	0.13	0.16	0.02
Jazz	0.06	0.12	0.10	0.04	0.00	**0.54**	0.01	0.04	0.02	0.07
Metal	0.06	0.00	0.01	0.08	0.02	0.00	**0.73**	0.00	0.01	0.10
Pop	0.00	0.00	0.07	0.08	0.06	0.02	0.00	**0.68**	0.08	0.03
Reggae	0.05	0.00	0.07	0.05	0.07	0.01	0.01	0.07	**0.62**	0.05
Rock	0.06	0.00	0.19	0.09	0.04	0.10	0.11	0.05	0.05	**0.30**

3. EXPERIMENTAL RESULTS

The GTZAN dataset [6] was used to evaluate the performance of the algorithm. It consists of 1000 songs of 10 different genres, 100 from each genre: blues, classic, country, disco, hiphop, jazz, metal, pop, reggae and rock. The success of classification was evaluated using 10-fold cross validation. The training set was chosen randomly once, and the testing set was chosen randomly 10000 times. The results presented here are the average results for the 10000 testing sets.

First, in order to evaluate the feasibility of the algorithm, we classified two distinct musical genres from GTZAN dataset - metal and classic (100 songs from each genre). The accuracy of classification when using the diffusion maps coordinates was $96.74 \pm 3.75\%$ (mean and standard deviation), and when using the feature-vectors themselves, without mapping with diffusion maps, it was only $87.99 \pm 6.83\%$. This means that the mapping contributes to better classification results.

Then, we tried to classify the whole data set (10 different genres). The accuracy of classification when using the diffusion maps coordinates was $56.55 \pm 4.50\%$ (mean and standard deviation), and when using the feature vectors themselves, it was $28.27 \pm 3.93\%$. In both cases there is an improvement when mapping the feature-vectors to the lower dimensional manifold before the classification.

The confusion matrix for classification using the diffusion maps coordinates is presented in Table 1. The names of genres without quotation marks represent the true genres, and those with the quotation marks represent the genres which the songs were classified to.

From Table 1 we see that the highest classification percentages are given to the correct genre in all cases.

Next, we examined the algorithm on 5 genres only - blues, classical, metal, pop, reggae. The accuracy of classification when using the diffusion maps coordinates was $84.91 \pm 4.88\%$, and when using the feature vectors themselves, it was $49.89 \pm 6.21\%$. The confusion matrix for the classification using the diffusion maps coordinates is presented in Table 2.

Here the classification results are much better, and they are significantly better when using the diffusion maps coordinates rather than the feature vectors.

The next experiment was to cluster the songs into pairs of genres - blues & country, classical & jazz, metal & rock, pop & hiphop and disco & reggae. The accuracy of classification when using the diffusion maps coordinates was $64.88 \pm 4.29\%$, and when using the feature vectors themselves, it was $43.63 \pm 4.33\%$. The confusion matrix is presented in Table 3.

Table 2: Averaged Confusion Matrix Using Diffusion Coordinates - 5 Genres

	"Blues"	"Classic"	"Metal"	"Pop"	"Reggae"
Blues	**0.85**	0.03	0.09	0.00	0.03
Classic	0.04	**0.91**	0.02	0.01	0.02
Metal	0.11	0.00	**0.88**	0.01	0.00
Pop	0.00	0.01	0.01	**0.88**	0.10
Reggae	0.12	0.01	0.01	0.13	**0.73**

Table 3: Averaged Confusion Matrix Using the Diffusion Coordinates - After Clustering to Pairs

	"Blues & Country"	"Classical & Jazz"	"Metal & Rock"	"Pop & Hiphop"	"Disco & Reggae"
Blues & Country	**0.64**	0.08	0.14	0.04	0.10
Classical & Jazz	0.12	**0.75**	0.07	0.03	0.03
Metal & Rock	0.14	0.06	**0.63**	0.05	0.12
Pop & Hiphop	0.06	0.01	0.05	**0.69**	0.19
Disco & Reggae	0.12	0.02	0.13	0.20	**0.53**

It is important to demonstrate the significant advantage of this method over Principal Component Analysis (PCA) as a method for dimensionality reduction, as indicated in Section 2. Because of the global and linear nature of PCA we would expect its classification results to be inferior. We performed dimensionality reduction using PCA to the same dimension as before (10), and received accuracy results (when classifying 10 different genres) of $28.28 \pm 3.91\%$, the same as when using the original 19 feature vectors for classification. This means that the 10 first principal components captured the important information of the feature vectors, but did not add any information that would improve the classification results, unlike the diffusion map, which improved the classification results.

4. CONCLUSIONS

Using the method of "Diffusion maps" for manifold learning leads to improved classification of music by genre and to reduction of the dimension of the problem. From this work it seems that the features that distinguish between different genres lie in a non-linear, lower-dimensional manifold, and therefore the classification of the music signals should be conducted in this manifold, and not in the original space of features. Moreover, we achieve the advantage of dimensionality reduction, which leads to lower computational complexity and saves storage space.

Future work may include comparison to other methods of manifold learning, such as ISOMAP, LLE, Laplacian Eigenmaps or Hessian Eigenmaps.

REFERENCES

[1] R. Coifman and S. Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21:5–30, June 2006.

[2] R. Coifman and S. Lafon. Geometric harmonics: A novel tool for multiscale out-of-sample extension of empirical functions. *Applied and Computational Harmonic Analysis*, 21:31–52, June 2006.

[3] S. Lafon, Y. Keller, and R. R. Coifman. Data fusion and multicue data matching by diffusion maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(11):1784–1797, Nov. 2006.

[4] C. McKay and I. Fujinaga. Automatic genre classification using large high-level musical feature sets. In *Proceedings of the International Conference on Music Information Retrieval*, pages 525–530, Barcelona, Spain, Oct. 2004.

[5] J. Shlens. A tutorial on principal component analysis. Technical report, Center for Neural Science, New York University, New York City, Apr. 2009.

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

[7] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian. Musical genre classification using support vector machines. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, volume 5, pages 429–32, Apr. 2003.