Unsupervised Music Motifs Extraction

Oriol Nieto
N15455073
Music and Audio Research Laboratory (MARL)
Final Project for Machine Learning
New York University
December 2010
Email: onc202@nyu.edu

***Abstract*—We propose two different unsupervised methodologies to extract Music Motifs out of annotated music data. The techniques described are the Convolutional Sparse Coding (CSC) and the Shift Invariant-Probabilistic Latent Component Analysis (SI-PLCA). We propose a structure for generating a meta-chromagram out of annotated music data to input to the different learning techniques. This proposed meta-chromagram structure stores all the relevant information to be able to reconstruct the original annotated music data out of it in a lossless process. We discuss why we should need the probabilistic component in the CSC in order to produce good results.**

I. Introduction

Music Motifs in music are the shortest music ideas present in a song. They can describe the song up to some extent, since they are recurring figures that can take different shapes in sound (*i.e.* different instruments can produce them at different times), different tempos or different keys [4]. We want to be able to extract the Music Motifs out of a song to be able to understand better the patterns and the structure of the whole song, since this would be the smallest analyzable element of it [7].

Different approaches to extract repetition in music or extract the most representative part of a song (this technique is called *Audio Thumbnailing*) have been developed ([9], [8], [2]). These techniques are focused on audio signals rather than annotated music data. In this work we propose a structure for generating a meta-chromagram out of an annotated music data (in our case we use annotated music data encoded using MIDI) to be able to learn the different Music Motifs out of it, following unsupervised techniques. The meta-chromagram stores all the relevant information in a 4 dimensional structure so that the process to reconstruct the annotated music data out of it is a lossless process.

The used techniques to proceed with this approach are Convolutional Sparse Coding (CSC [1]) and Shift-Invariant Probabilistic Latent Component Analysis (SI-PLCA [3]). This last technique is a probabilistic approach that comes from the Sparse Convolutive Non-Negative Matrix Factorization [2].

For the Sparse Coding technique, we implemented the *Iterative Shrinkage-Thresholding Algorithm* (ISTA [14]) and the *Fast Iterative Shrinkage-Thresholding Algorithm* (FISTA [6]). Both implementations accept the meta-crhomagram structure to produce the Musical Motifs results. Although perfect reconstruction using Sparse Coding could be produced by learning their respective dictionaries using these two algorithms, no Musical Motifs could be successfully extracted. We discuss why we may need probabilistic information in order to predict the motifs correctly.

For the SI-PLCA technique, we modified the available code[1] from the publication [2]. With our modifications, this algorithm accepts the meta-crhomagram input, and evaluates the song for Music Motifs instead of general repetition or most representative part of the song.

Outline: This document is structured as follows. In Section II the methodology of the two techniques are described. Also the proposed meta-chromagram structure is described. In Section III the Results of this project are given. The Section IV describes the Source Code, how it is structured and how to run it. Finally, section V gives the conclusions of our proposed approaches to extract Music Motifs using unsupervised techniques and comments the possible future work.

II. Methodology

In this Section we describe the meta-crhomagram structure that will be used as an input for the different techniques to approach Musical Motifs extraction. Then, we describe and discuss these two different techniques.

A. Meta-Chromagram Structure

The chromagram is defined as the restructuring of a spectral representation in which the frequencies are mapped onto a limited set of 12 chroma values in a many-to-one fashion [15]. These 12 different chroma values represent each of the 12 different pitches that are present in the western music. We map the energy to these different pitches across time, so the chromagram C is a 3 dimensional structure (pitch $F \in 1 \dots 12$ $\times$ energy E $\times$ time length L):

$$C \in \Re^{F \times E \times L} \tag{1}$$

An example of chromagram from annotated musical data can be found in Figure 1.

[1] http://marl.smusic.nyu.edu/resources/siplca-segmentation

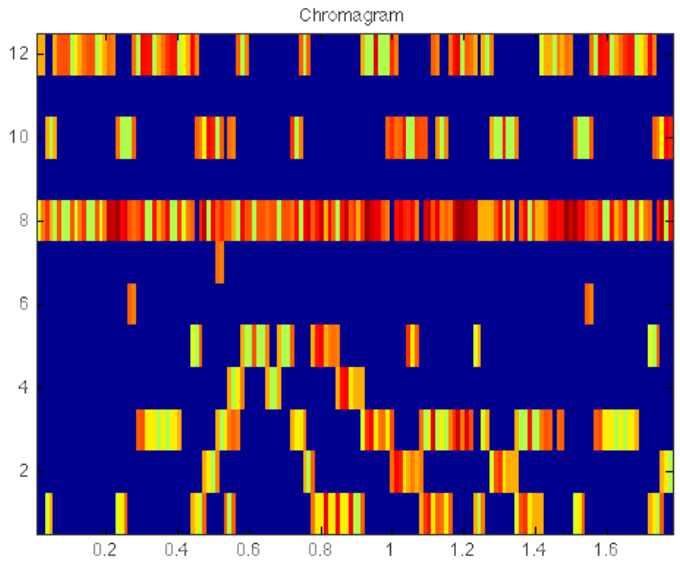

Fig. 1. Example of Chromagram from an annotated musical data of the *Blackbird* song by *The Beatles*

We propose to add meta-data information to this chromagram, by taking the advantage of working with annotated music data. We call this new structure *meta-chromagram*. A new dimension N is included, where, for each note added in the chromagram, we add a new array of notes, including pitch, velocity and type of instrument for each note. This information is known since we are working with annotated music data (we are using the MIDI encoding for this purpose). This array of notes will be of dimension n, where n is the number of notes that are played simultaneously in that precise moment in time with the same chroma. n is going to be of variable length across the whole meta-chromagram. The meta-chromagram K is then defined as follows:

$$K \in \Re^{F \times E \times L \times N} \tag{2}$$

So the meta-chromagram is a 4 dimensional structure that makes the process of going from the meta-chromagram to the annotated music data a lossless process.

We use this structure as the input data for our techniques described below. The techniques will use only the first 3 dimensions (*i.e* a traditional chromagram C) to find the Music Motifs, and will use the last dimension to reconstruct the annotated musical data.

The implementation to create a meta-chromagram from an annotated music data (*i.e.* a MIDI file) can be found in the source code for this project. It is written in C++ and it uses the library JDKSMIDI[2], an open source library to read and write MIDI files in C++.

To have instrument invariance for recognizing all music motifs, our meta-chromagrams will be constructed instrument by instrument sequentially instead of having all the instruments playing at once, having a long meta-chromagram that will be the length of the song L times the number of instruments S. This will help us finding all the motifs without having to worry whether another instrument is playing at the same time a different type of motif when the motif to be detected is played by another instrument. We added 5 seconds silence between instruments so that the algorithm could detect the change of instruments.

In our project, we use 10ms windows in order to create the meta-chromagrams. This amount of time gives us enough resolution for our problem.

B. Convolutional Sparse Coding

The first described approach to extract Music Motifs using unsupervised techniques is Convolutional Sparse Coding (CSC) [1]. Even though our techniques input a meta-chromagram structure to be able to reconstruct the Music Motifs as annotated music data, we will simplify here the problem by assuming the input is a regular chromagram C (extended times the number of instruments, to have instrument invariance as mentioned above).

The main problem is to reconstruct the chromagram C by using the dictionary W and the activation weights H:

$$C \approx W * H \tag{3}$$

Where W would be a 3 dimensional matrix $F \times L \times K$ that represent all different Musical Motifs as short-time length chromagrams, where K is the number of different Musical Motifs found. Then, the activation matrix H activates the different motifs during a given time.

Using CSC, the energy function to approach this problem would be:

$$E(C, W, H) = \frac{1}{2}||C - W * H||_2^2 + \lambda||H||_1 \tag{4}$$

Where λ is the *sparse coefficient* in H. We want H to be sparse enough to be able to obtain the best Musical Motifs representation. Basically, the sparsity will be a prior of H.

The equation 4 is similar to a type of equations called Basis Pursuit Denoising and it can be solved using the *Iterative Shrinkage-Thresholding Algorithm* (ISTA [14]) or the *Fast Iterative Shrinkage-Thresholding Algorithm* (FISTA [6]).

We implemented both in Matlab by applying an Expectation-Maximization (EM) algorithm to learn the dictionary W. The source code is available in the package of this project.

1) E-step: In this step we will fix W and will solve using a Sparse Coding algorithm to get H (both ISTA and FISTA algorithms will work). We implemented both the ISTA and FISTA algorithms with a constant step size L as described in [6].

Using these algorithms, H would be:

$$H_{k+1} = \tau_{\lambda/L}(H_k - W^T(C - W * H)) \tag{5}$$

[2]https://github.com/jdkoftinoff/jdksmidi

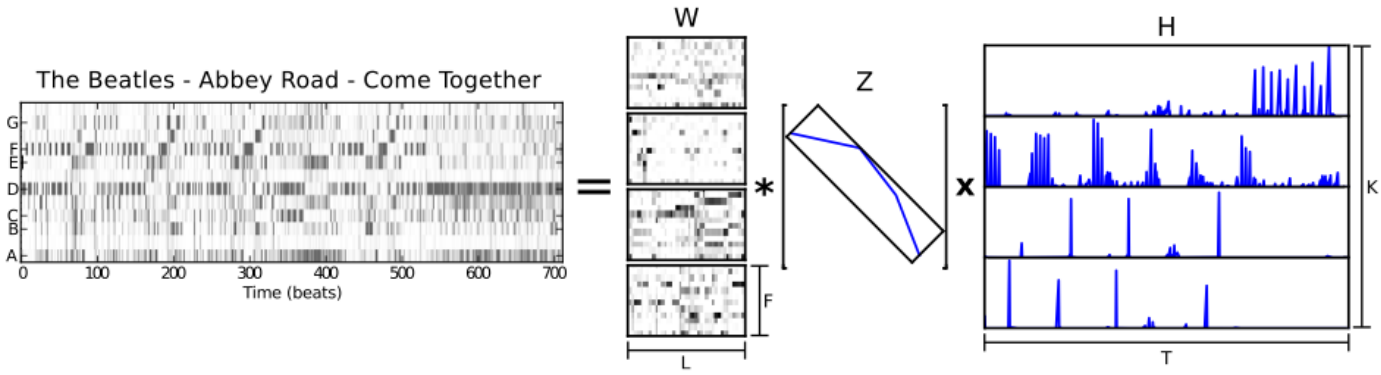

Fig. 2. Example of the SI-PLCA technique, having $K = 4$ and $L = 60$. Taken from [2]

where k is the iteration number and τ_a is the shrinkage function defined as:

$$\tau_a(x)_i = (|x|_i - a)sign(x_i) \tag{6}$$

To find the step size L we search for an L such as $Q(H_k, H_{k-1}) > F(H_k) + G(H_k)$, where $Q(z, z')$ is the quadratic approximation:

$$Q(z, z') = F(z') + \langle z - z', \nabla F(z') \rangle + \frac{L}{2}||z - z'||^2 + G(z) \tag{7}$$

and $F(H)$ and $G(H)$ are the two parts of equation 4:

$$F(H) = \frac{1}{2}||C - W * H||_2^2 \tag{8}$$

$$G(H) = \lambda||H||_1 \tag{9}$$

We repeat this until convergence. Once we get H we are able to find the dictionary W in the *M-Step*.

2) M-step: In this step we fix H (updated from the *E-Step*). We want to learn the dictionary W, and to do so, we apply an online stochastic gradient descent algorithm:

$$W_{k+1} = W_k - \eta \frac{\delta E(C, W, H)}{\delta W} \tag{10}$$

where η is the *learning rate* coefficient.

Once we have learnt W and H using this EM algorithm described above, we can apply a *K-means* algorithm through $|H|$ to sort the most relevant Music Motifs (*i.e.* the ones that repeat the most) so that we can give an order and detect the most relevant ones.

In order to have pitch invariant, we must have 12 different versions of each dictionary W, one for each pitch. This is a drawback in terms of computational complexity with respect to the other proposed technique we are about to describe.

C. Shift-Invariant Probabilistic Latent Component Analysis

The second technique to approach our problem of extracting Music Motifs out of annotated music data is called *Shift-Invariant Probabilistic Latent Component Analysis* (SI-PLCA) and is described in [2] and [3].

The idea is to extend the Non-Negative Matrix Factorization problem (NMF) to add a probabilistic component. Using PLCA, each column of W and each row of H is represented as a multinomial probability distribution and also adds additional distribution to each basis function, called mixing weights (a diagonal matrix Z). The formulation would be:

$$C \approx WZH = \sum_{k=0}^{K-1} w_k z_k h_k^T \tag{11}$$

The Shift-Invariance is obtained by using convolution and making W a 3 dimensional matrix $F \times L \times K$, such that:

$$C \approx \sum_{k=0}^{K-1} W_k * z_k h_k^T \tag{12}$$

An example extracted from [2] can be found in Figure 2.

The implementation for this technique is written in Python using an EM algorithm explained in the original paper [2] and a reference to the source code is found in the same document. We modify it in order to accept our special type of meta-chromagrams.

III. Results

The results obtained by using CSC were not successful, since we could not figure an optimal way to represent the dictionaries W as representation of Music Motifs. However, we were able reconstruct our original chromagrams using CSC (thus, proving our implementations for ISTA and FISTA were successfully working).

We are considering adding a probabilistic component to it, in order to produce good results in the future.

On the other hand, the modified SI-PLCA algorithm managed to produce good results. The song *Blackbird* by *The Beatles* was used to evaluate the technique, and different motifs could be successfully extracted. By reconstructing the annotated music data (a lossless process since we're using the meta-chromagrams, described in Section II), we could hear clearly the different motifs and successfuly claim they are the key parts of the song.

The motifs extracted were around 10 seconds long, and even though that changes depending on the song, we believe the technique could be improved in order to obtain shorter motifs.

IV. Source Code

The Source Code[3] is structured as follows: The file **MotifsXtraction.zip** contains a directory called **MotifsXtraction**. This directory contains all the source code for this project.

A. Meta-Chromagram Implementation

The Meta-Chromagram implementation is found in the root directory of our project. The implementation is written in C++, and it contains a project file to open it conveniently with XCode (OSX Programming IDE). This project file is called **Chromagram.xcodeproj**. It links our code with the libraries of JDSKMIDI.

The main file to run it is **main.cpp** and it gets one argument, which is a MIDI file. MIDI files are found in the subdirectory called **midi**. By default, if run with XCode, the Chromagram will take the file **Blackbird.mid** as the input to the algorithm.

The output of the algorithm is placed in the the subdirectory **siplca-segmentation**, in the form of two different files: **feats.txt** and **beats.txt**. Both of these files create the chromagram.

B. CSC Implementation

The Convolutional Sparse Coding implementation can be found in the subdirectory called **CSC**. These are a series of Matlab files, where the main file is found in the file called **csc.m**.

The Sparse Coding algorithms are found in files **ISTA.m** and **FISTA.m**, and they implement the ISTA and FISTA algorithms respectively.

For simplicity, when **csc.m** is run, it evaluates a simple array V and learns the dictionaries W and H that compose V. One could run the CSC with a Chromagram by inputing the output of the Meta-Chromagram project by hand.

C. SI-PLCA Implementation

The Shift-Invariance Probabilistic Latent Component Analysis implementation is found in the subdirectory called **siplca-segmentation**. It is written in Python by Ron Weiss, and we modified it to accept our chromagrams, and to extract shorter motifs.

To run it, type **./motifs_xtraction.sh output.txt** from the siplca-segmentation subdirectory, and it will write the output of the algorithm to the **output.txt** file. The algorithm takes the input of the output of the Meta-Chromagram Implementation. So to test a different file (by default is the chromagram generated from the **Blackbird.mid** file), one must run the Meta-Chromagram algorithm with another input file (*e.g.* **furelise.mid**).

[3]Found here: http://ccrma.stanford.edu/∼urinieto/research/MotifsXtraction.zip

V. Conclusions

Two approaches to extract Music Motifs have been discussed in this document: Convolutional Sparse Coding and Shift-Invariance Probabilistic Latent Analysis Component. A structure called meta-chromagram has been presented, which gives a lossless process from a meta-chromagram to an annotated music data representation (such as MIDI), by keeping the simplicity of a chromagram structure.

The CSC technique was able to reconstruct the chromagram successfully, by learning the correct dictionaries and weights. However, Music Motifs were not able to be extracted from the dictionary. We believe that we need to add probabilistic information to the dictionary and/or to the weights to obtain the desired results. Further work should be focused on finding the way to add a probabilistic component to it to obtain better results.

The SI-PLCA methodology was able to find Music Motifs from a meta-chromagram generated out of an annotated music data. The results are promising, even though shorter Music Motifs would be more desirable.

By extracting Music Motifs in an unsupervised way, we are closer to analyze in an easier way annotated music data, and thus being able to understand the underneath musical patterns and structure within the context of a song.

Acknowledgment

The author would like to thank Mary Farbood, Koray Kavukcuoglu, Yann LeCun, Juan Pablo Bello and Ron Weiss for their invaluable help for making this project.

References

[1] K. Kavukcuoglu, P. Sermanet, Y.L Boureau, K. Gregor, M. Mathieu, and Y. LeCun, *Learning Convolutional Feature Hierachies for Visual Recognition.* in Advances in Neural Information Processing Systems (NIPS), 2010.

[2] R.J. Weis and J.P. Bello, *Identifying Repeated Patterns in Music Using Sparse Convolutive Non-Negative Matrix Factorization.* In Proc. ISMIR, 2010.

[3] R.J. Weiss and J.P. Bello, *Unsupervised Discovery of Temporal Structure in Music*, yet to be published.

[4] A. Ockelford, *Repetition in music: theoretical and metatheoretical perspectives.* Volume 13 of Royal Musical Association monographs. Ashgate Publishing, 2005.

[5] M.A.T. Figueiredo, R.D. Nowak and S.J. Wright, *Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems*, IEEE J. Sel. Top. Signal Process., 1, pp. 586597, 2007.

[6] A. Beck and M. Teboulle, *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems*, SIAM J. IMAGING SCIENCES, Society for Industrial and Applied Mathematics Vol. 2, No. 1, pp. 18320, 2009.

[7] J.J. Nattiez, *Music and Discourse: Toward a Semiology of Music (Musicologie gnrale et smiologue, 1987)*. Translated by Carolyn Abbate. Princeton, NJ: Princeton University Press. ISBN 0691091366/ISBN 0691027145, 1990.

[8] M.A. Bartsch and G.H. Wakefield, *Audio thumbnailing of popular music using chroma-based representations.* IEEE Trans. Multimedia, 7(1):96104, 2005.

[9] J.P. Bello. *Grouping recorded music by structural similarity.* In Proc. ISMIR, pages 531536, 2009.

[10] M. Casey and M. Slaney. *Song Intersection by Approximate Nearest Neighbor Search.* In Proc. ISMIR, 2006.

[11] D.P.W. Ellis and G.E. Poliner. *Identifying cover songs with chroma features and dynamic programming beat tracking.* In Proc. ICASSP, pages IV14291432, 2007.

[12] M. Levy and M. Sandler. *Structural Segmentation of Musical Audio by Constrained Clustering.* IEEE Trans. Audio, Speech, and Language Processing, 16(2), 2008.

[13] B. A Olshausen and D.J. Field *Sparse coding with an overcomplete basis set: a strategy employed by v1?* Vision Research, 37(23):33113325, 1997.

[14] A. Chambolle, R.A. DeVore, N.Y. Lee and B.J. Lucier, *Nonlinear wavelet image processing: Variational problems, compression, and noise removal through wavelet shrinkage*, IEEE Trans. Image Process., 7, pp. 319335. 1998.

[15] S. Pauws, *Musical key extraction from audio.* In Proceedings of the International Society for Music Information Retrieval (ISMIR), 2004.