CONVEX NON-NEGATIVE MATRIX FACTORIZATION FOR AUTOMATIC MUSIC STRUCTURE IDENTIFICATION

Oriol Nieto[*]

Music and Audio Research Lab
New York University
oriol@nyu.edu

Tristan Jehan

The Echo Nest
tristan@echonest.com

ABSTRACT

We propose a novel and fast approach to discover structure in western popular music by using a specific type of matrix factorization that adds a convex constrain to obtain a decomposition that can be interpreted as a set of weighted cluster centroids. We show that these centroids capture the different sections of a musical piece (e.g. verse, chorus) in a more consistent and efficient way than classic non-negative matrix factorization. This technique is capable of identifying the boundaries of the sections and then grouping them into different clusters. Additionally, we evaluate this method on two different datasets and show that it is competitive compared to other music segmentation techniques, outperforming other matrix factorization methods.

Index Terms— matrix factorization, music structure analysis, segmentation

1. INTRODUCTION

Identifying music structure in an automated fashion is a common task in systems that manage any type of music information, especially those containing large collections of songs. The automatic identification of structure in music is a topic that has been widely investigated in the music informatics research community [1]. The main goal is to segment a piece in its different sections (e.g. verse, chorus), a task that is often divided into two different subproblems: (i) the finding of the boundaries that separate the sections and (ii) the clustering (or labeling) of these sections into different groups based on their similarities.

The classic approach to identify boundaries is to apply a "checkerboard" kernel over the diagonal of a Self Similarity Matrix (SSM) of certain —commonly beat-synchronous— features, thus obtaining a novelty curve from which to extract the boundaries by extracting its more prominent peaks [2, 3, 4]. The size of this kernel defines the amount of previous and future features being taken into account. Other approaches include the usage of supervised learning [5] or variants of SSM also known as lag matrices [6]. As for the grouping subtask, it can be viewed as an audio similarity problem. Different methods have been proposed: using Gaussian Mixture Models (GMM) [7], a variant of Nearest Neighbor Search (NNS) [8], and Non-negative Matrix Factorization (NMF) [9]. Finally, other methods combine both tasks into one sole algorithm, e.g. using Hidden Markov Models (HMM) [10, 11], a probabilistic version of convolutive NMF [12], or k-means clustering [13].

Our approach is based on the NMF method proposed in [9], which we extend by adding a convex constrain [14] that results in weighted cluster centroids that represent the different sections of a musical piece in a more effective manner. Moreover, we show that it is possible to efficiently extract music boundaries by clustering the decomposition matrices, which take into account the repeated parts across the song instead of just detecting sudden local changes. Therefore, the proposed algorithm aims to address the two main subtasks of music segmentation —i.e. finding boundaries and clustering sections.

2. FEATURE EXTRACTION

2.1. Beat-Synchronous Chromagram

In this work we make use of the chroma features described in [15] that are provided by the Echo Nest API[1].

A chroma feature is characterized by a 12-dimensional vector that represents the amount of energy that can be found in each of the 12 different pitches that commonly exist in the western popular music folded into one single octave. This is achieved by applying a constant-Q transform across the entire spectrogram and then folding it into one octave comprising the 12 quantized pitches. When these features are stack together following the song structure in a $N \times 12$ matrix, we generate a so-called *chromagram*, where N is the number of time frames in which the musical piece has been divided.

Moreover, we resample onset-based asynchronous chroma features (as found through The Echo Nest *track* API) to beats,

[*]Thanks to *Fundación Caja Madrid* for the funding

[1]The Echo Nest Analyzer API, *http://developer.echonest.com*

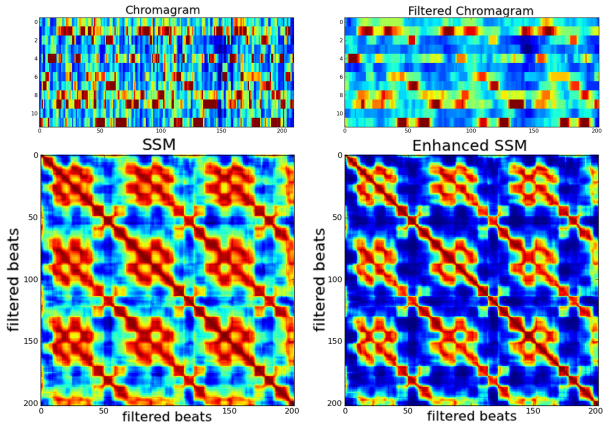

Fig. 1. Example of a chromagram (top-left), a pre-filtered chromagram with $h = 9$ (top-right), an original SSM using the correlation distance (bottom-left), and an enhanced SSM (bottom-right), of the song Help! by The Beatles.

thus reducing greatly the number of frames across the musical piece, and leading to *beat-synchronous* chromagrams.

2.2. Pre-Filtering and SSM Enhancement

A series of transformations are applied to the chromagram in order to better distinguish the different parts of a song (as it is common in this type of problem [1]). First, a sliding median filter of size h is run against each of the beat-synchronous chromagram channels. The median filter gives sharper edges than a regular mean filter, which is useful in obtaining section boundary precision. By filtering features across time, we retain the most prominent chromas within the h-size window and remove smaller artifacts, which are irrelevant in our context. In Figure 1 we show the example of a non-filtered and its corresponding pre-filtered chromagram.

We then compute the SSM of the pre-filtered beat-synchronous chromagram. The SSM gives us pair-wise comparisons of a given set of features using a specific distance measure and stores the results in an $N \times N$ symmetric matrix D, such that $D(i, j)$ holds the distance between the features of the beat indices i and j. In this case $D(i, j) = D(j, i)$ and $D(i, i) = 0$. It is essentially a specific instance of the more generic recurrence plots [16], but using distances (or similarities) instead of binary values. In our experiments we found that the *Correlation* distance gave better results than other distances, including the Euclidean, Cosine or Manhattan distance. The correlation distance is defined as follows:

$$d(\mathbf{x}, \mathbf{y}) = \frac{(\mathbf{x} - \mu_{\mathbf{x}}) \cdot (\mathbf{y} - \mu_{\mathbf{y}})}{||\mathbf{x} - \mu_{\mathbf{x}}||_2 ||\mathbf{y} - \mu_{\mathbf{y}}||_2} \quad (1)$$

where $||\cdot||_2$ stands for the Euclidean distance, $\mu_{\mathbf{x}}$ denotes the mean of the feature vector $\mathbf{x}$, and $\cdot$ represents the dot product.

Finally, we enhance the SSM by using a power-law expansion (using the power 2 empirically gave us the best results), such that close similarities will be closer and distant similarities will be more distant. This improves the contrast of the SSM and results in clearer matrix factorizations. After the exponentiation, the final step consists of normalizing the entire matrix between 0 (very dissimilar) and 1 (equal). We illustrate this enhancement in Figure 1.

3. CONVEX NMF IN MUSIC SEGMENTATION

3.1. Convex NMF Description

The factorization of an input feature matrix $X \in \mathbb{R}^{N \times p}$, composed of $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$, which has N row observations $\mathbf{x}_i$ of p features, can be described as $X \approx FG$, where $F \in \mathbb{R}^{N \times r}$ can be interpreted as a cluster row matrix, $G \in \mathbb{R}^{r \times p}$ is composed of the indicators of these clusters, and r is the rank of decomposition. In NMF, both F and G are enforced to be positive (i.e. X must be positive too). We denote by $\mathbf{z}$ a row vector and by $\mathbf{z}^T$ a column one.

C-NMF adds a constrain to $F = (\mathbf{f}_1^T, \ldots, \mathbf{f}_r^T)$ such that its columns $\mathbf{f}_j^T$ become convex combinations of the features of X:

$$\mathbf{f}_j^T = \mathbf{x}_1^T w_{1j} + \ldots + \mathbf{x}_p^T w_{pj} = X \mathbf{w}_j^T \quad j \in [1 : r] \quad (2)$$

For a linear combination to be convex, all coefficients w_{ij} must be positive and the sum of each set of coefficients $\mathbf{w}_j^T$ must be 1. Formally: $w_{ij} \geq 0, \sum_j w_{ij} = 1$.

This results in $F = XW$, where $W \in \mathbb{R}^{p \times r}$, which makes the rows $\mathbf{f}_i$ interpretable as weighted cluster *centroids*, representing, in our case, better sections of the musical piece as we will see in subsection 3.3 when computing the decomposition matrices. The decomposition matrices R_j are obtained as follows: $R_j = \mathbf{f}_j^T \mathbf{g}_j$, where $j \in [1 : r]$. Finally, C-NMF can be formally characterized as: $X \approx XWG$.

For a more detailed description of C-NMF with an algorithm explanation and sparsity discussion we refer the reader to [14]. A good review of algorithms for NMF can be found in [17]. Lastly, a good example of C-NMF in computer vision can be found in [18].

3.2. C-NMF vs NMF

As opposed to NMF, in C-NMF the matrix F is a set of convex combinations of the rows of the input matrix X (see equation 2). Since, in our case, X is a SSM, we have, for each row $\mathbf{x}_i$, the similarity of the time frame i with the rest of the time frames. Thus, each row $\mathbf{f}_i$ stores information about the time frame i across the entire song too. That is why, as Figure 2 shows, the boundaries become much clearer in the decomposition matrices when interpreted as matrices of row-vector features.

Another important benefit for our application of music segmentation is that the matrices W and G are naturally sparse when adding this convex constrain, as opposed to traditional NMF (where G is not necessarily sparse). This

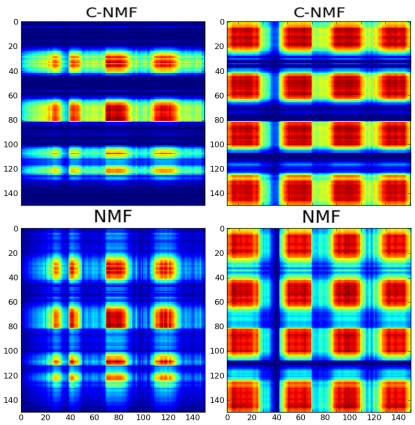

Fig. 2. Decomposition matrices of C-NMF (top) and NMF (bottom) with a rank of decomposition of $r = 2$ from the song Tell Me Why by The Beatles.

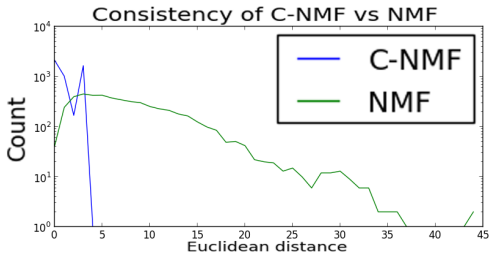

Fig. 3. Logarithmic histogram of distances between 100 sets of decomposition matrices obtained with C-NMF (blue) and NMF (green) from the song Help! by The Beatles.

results in C-NMF being more likely to find similar decomposition matrices for the same input than NMF, which is more sensitive to its initialization. To illustrate this we execute both C-NMF and NMF $T = 100$ times for the same song with $r = 2$. We compute the pair-wise difference $C(M^i, M^j)$ between their resulting sets of decomposition matrices $M^n = \{R_1^n, \ldots, R_r^n\}$ (where n is the execution index, $n \in [1:T]$). This is formally illustrated in Equation 3.

$$C(M^i, M^j) = \sum_{m=1}^{r} ||{M_m}^i - {M_m}^j||_2 \quad i, j \in [1:T] \tag{3}$$

In Figure 3 we plot the logarithmic histogram of these differences for each method, so that the shorter the difference, the more consistent the technique will be. As can be seen, C-NMF's greatest difference is smaller than 5, and NMF's greatest difference is almost 45, therefore C-NMF is more consistent than NMF.

3.3. Applying C-NMF in Music Segmentation

In this subsection we describe how C-NMF can be useful in the task of music structure analysis. We divide this part into the two main problems of music segmentation: boundaries and clustering.

3.3.1. Finding Boundaries

We run k-means clustering with $k = 2$ to each one of the C-NMF decomposition matrices, interpreting them as row-vector features. We efficiently obtain the section boundaries that best divide each section of the matrices by not only looking at local similarities but also the global song structure due to the properties of the SSM. The choice of $k = 2$ allows us to detect boundaries (i.e. there's a boundary or not), regardless of how the various sections cluster.

One computational advantage of applying k-means clustering to an NMF (either convex or not) decomposition matrix is that, when whitening the data (i.e. making it unit variance), due to the fact that we use a SSM as an input and the similarities between NMF and k-means clustering [19], we obtain a one-dimensional feature array of observations (i.e. all rows become equal), which makes the computational process cheaper. Once we have boundaries for each matrix, we combine them within a distance window of size w so that boundaries close to each other get merged in their average location.

3.3.2. Clustering Sections

The main idea is to use the diagonals of the C-NMF decomposition matrices to form a new feature space from which to cluster the different sections, as described in [9], using the previously found boundaries. In this work we make use of the Euclidean distance for clustering, and put the exploration of different distance measures like the Bayesian Information Criterion (BIC) or the Mahalanobis aside for future work. The main drawback of this method is to decide on the number of sections K, which is used to cluster the new feature space and it is a highly sensitive parameter to the musical style of the dataset.

4. EVALUATION

We evaluate our algorithm with the annotated Beatles dataset[2] corrected by the Tampere University of Technology (TUT Beatles)[3]. That dataset is composed of 176 songs and is traditionally used to evaluate such segmentation task [10, 4, 9, 12]. We also evaluate against the Internet Archive part of the more recent SALAMI dataset [20], which contains 253 freely available songs.

We used the following parameters in our evaluation: $h = 9$ beats for the size of the median-filter window, $w = 8$ beats for the size of the window that merges boundaries, $r = 2$ for the number of decomposition matrices, and $K = 4$ for the number of section types per song. We leave an exhaustive exploration of the parameters for future work due to the limitation of space, while still showing that we can obtain good results with this set of arguments.

[2] http://www.icce.rug.nl/∼soundscapes/DATABASES/AWP/awp-notes_on.shtml

[3] http://www.cs.tut.fi/sgn/arg/paulus/structure.html

TUT Beatles Dataset								
	Clustering					Boundaries		
Method	F	P	R	S_o	S_u	F	P	R
C-NMF	**59.3**	48.9	83.2	49.8	47.8	57.3	54.9	64.6
NMF	56.6	48.8	77.7	43.7	49.6	**58.9**	54.7	67.7
SI-PLCA	55.8	46.3	80.7	41.0	50.6	23.2	50.9	17.2
Kaiser[9]	60.8	61.5	64.6	–	–	50.0	46.5	52.2
SALAMI (Internet Archive) Dataset								
	Clustering					Boundaries		
Method	F	P	R	S_o	S_u	F	P	R
C-NMF	**53.1**	44.0	81.0	50.6	44.3	45.1	43.0	52.3
NMF	51.5	42.8	77.6	37.9	45.6	**48.8**	44.0	62.7
SI-PLCA	51.3	55.8	52.1	44.2	51.4	24.8	45.1	18.4

Table 1. Results for three different algorithms (C-NMF, NMF, and SI-PLCA) applied to two different datasets: TUT Beatles (top) and the Internet Archive subset of SALAMI (bottom). The table shows the results for clustering (left) and boundaries (right).

The results of the algorithm are compared against two other techniques that use matrix factorization for music segmentation: SI-PLCA [12] and a variant of our algorithm that uses classic NMF instead of C-NMF. The parameters used for SI-PLCA are the ones proposed for MIREX (see source code[4]). The parameters used for NMF are identical to the ones used for C-NMF. The same features described in Section 2 were used for the three algorithms. Finally, we also compare the results for the TUT Beatles dataset with the ones reported in [9], obtained by using different chromas and the Mahalanobis distance for clustering.

4.1. Boundaries Evaluation

The boundaries are evaluated with the F-measure, which quantifies whether there is an estimated boundary within ± 3 seconds from the annotated one, as described in [21]. On the right side of Table 1 the F-measure with the precision (P) and recall (R) values are presented.

As is presented on the table, C-NMF and NMF outperform SI-PLCA in both the TUT Beatles and SALAMI datasets. NMF obtains slightly better results than C-NMF, however this could be due to the over segmentation of NMF when it happens to fall into a local minimum. Kaiser uses the traditional "checkerboard" technique [2] to obtain boundaries, and also gets a lower score than our proposed method of clustering the decomposition matrices, as described in subsection 3.3.1.

4.2. Clustering Evaluation

We evaluate the clustering task using the pairwise F-measure as explained in [10], with the addition of the entropy scores for over-segmentation (S_o) and under-segmentation (S_u), as suggested in [22]. The results are showed on the left side of Table 1.

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

C-NMF outperforms both NMF and SI-PLCA. We can see that S_u is slightly better for SI-PLCA, suggesting that both NMF and C-NMF under segment the data more than SI-PLCA. However, S_o indicates that SI-PLCA over-segments the data a bit more than the others. Kaiser method outperforms the rest, but we believe that by using other distances for clustering (like Mahalanobis, the one that Kaiser uses) we might obtain better results.

4.3. Discussion

This technique follows a stochastic process, so it is prone to fall into local minima. We experimentally found that a good number of iterations to run is around 30 for C-NMF and 100 for NMF, since, as we previously discussed in Section 3.2, C-NMF is more consistent. The features used in these experiments are not key-invariant, and it should be noted that adding key-invariance to the SSM, as described in [23], would improve the results (but also increase its running time).

A limitation of this technique is that it might not capture some boundaries originated from drastic changes in the features, as opposed to the "checkerboard" novelty curve technique. We believe that combining the most salient boundaries from both of these techniques could significantly improve the detection of boundaries, and would ultimately get us a better clustering of the sections.

C-NMF is considerably faster than SI-PLCA or the regular NMF because of the fewer number of iterations required. It would be interesting to formally compare the speed of each of these algorithms in the future, but it is already worth mentioning that SI-PLCA takes over 1000 seconds to run on the TUT Beatles dataset, while it only takes 170 seconds with the C-NMF approach. Computational efficiency is important when running this sort of algorithms over large datasets, as is the case for instance at The Echo Nest.

5. CONCLUSIONS

We introduced a new matrix factorization method to automatically identify the structure of a song. By adding a convex constrain to the NMF we showed that we obtain more consistent decomposition matrices, producing centroids that better represent the different sections of a song and improving their clustering (or labeling). Moreover, the method finds the boundaries of the sections by clustering the decomposition matrices. Our proposed algorithm was evaluated against the TUT Beatles and the SALAMI datasets, and we found better boundary and clustering results (by using NMF and C-NMF respectively) than other matrix factorization techniques while being computationally efficient. We discussed the limitations of our method, and proposed various ways of improving it.

6. REFERENCES

[1] Jouni Paulus, Meinard Müller, and Anssi Klapuri, "Audio-Based Music Structure Analysis," in *Proc of the 11th International Society of Music Information Retrieval*, Utrecht, Netherlands, 2010, pp. 625–636.

[2] Jonathan Foote, "Automatic Audio Segmentation Using a Measure Of Audio Novelty," in *Proc. of the IEEE International Conference of Multimedia and Expo*, New York City, NY, USA, 2000, pp. 452–455.

[3] Yu Shiu, Hong Jeong, and C.C. Jay Kuo, "Similarity Matrix Processing for Music Structure Analysis," in *Proc. of the 1st ACM workshop on Audio and Music Computing Multimedia*, Santa Barbara, CA, USA, 2006, pp. 69–76.

[4] Matthias Mauch, Katy Noland, and Simon Dixon, "Using Musical Structure to Enhance Automatic Chord Transcription," in *Proc. of the 10th International Society of Music Information Retrieval*, Kobe, Japan, 2009, pp. 231–236.

[5] Douglas Turnbull, Gert Lanckriet, Elias Pampalk, and Masataka Goto, "A Supervised Approach for Detecting Boundaries in Music Using Difference Features and Boosting," in *Proc. of the 5th International Society of Music Information Retrieval*, Vienna, Austria, 2007, pp. 42–49.

[6] Masataka Goto, "A chorus-section detecting method for musical audio signals," *Acoustics, Speech, and Signal Processing, 2003.*, vol. 2003, no. April, pp. 437–440, 2003.

[7] Ju-Chiang Wang, Hung-Shin Lee, Hsin-Min Wang, and Shyh-Kang Jeng, "Learning the Similarity of Audio Music in Bag-of-frames Representation from Tagged Music Data," in *Proc. of the 12th International Society of Music Information Retrieval*, Miami, FL, USA, 2011, pp. 85–90.

[8] Dominik Schnitzer, Arthur Flexer, Markus Schedl, and Gerhard Widmer, "Using Mutual Proximity to Improve Content-Based Audio Similarity," in *Proc of the 12th International Society of Music Information Retrieval*, Miami, FL, USA, 2011, pp. 79–84.

[9] Florian Kaiser and Thomas Sikora, "Music Structure Discovery in Popular Music Using Non-Negative Matrix Factorization," in *Proc. of the 11th International Society of Music Information Retrieval*, Utrecht, Netherlands, 2010, pp. 429–434.

[10] Mark Levy and Mark Sandler, "Structural segmentation of musical audio by constrained clustering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 2, pp. 318–326, Feb. 2008.

[11] Christopher Rhodes, Michael Casey, Samer Abdallah, and Mark Sandler, "A Markov-Chain Monte-Carlo Approach to Musical Audio Segmentation," in *Proc. of the IEEE International Conference on Acoustics Speech and Signal Processing*, 2006, pp. 797–800.

[12] Ron Weiss and Juan Pablo Bello, "Unsupervised Discovery of Temporal Structure in Music," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1240–1251, 2011.

[13] Geoffroy Peeters, Amaury La Burthe, and Xavier Rodet, "Toward Automatic Music Audio Summary Generation from Signal Analysis," in *Proc. of the 3rd International Society of Music Information Retrieval*, Paris, France, 2002, pp. 94–100.

[14] Chris Ding, Tao Li, and Michael I Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 1, pp. 45–55, Jan. 2010.

[15] Tristan Jehan, *Creating Music by Listening*, Ph.D. thesis, Massachusetts Institute of Technology, 2005.

[16] N Marwan, M Carmenromano, M Thiel, and J Kurths, "Recurrence Plots for the Analysis of Complex Systems," *Physics Reports*, vol. 438, no. 5-6, pp. 237–329, Jan. 2007.

[17] Daniel D Lee and H Sebastian Seung, "Algorithms for Non-negative Matrix Factorization," *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2000.

[18] Christian Thurau, Kristian Kersting, and Christian Bauckhage, "Convex Non-negative Matrix Factorization in the Wild," in *Proc. of the 9th IEEE International Conference on Data Mining*, Miami, FL, USA, Dec. 2009, pp. 523–532.

[19] Jingu Kim and Haesun Park, "Sparse Nonnegative Matrix Factorization for Clustering," Tech. Rep., GT-CSE-08-01, Georgia Institute of Technolgy, 2008.

[20] Jordan B. Smith, J. Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J. Stephen Downie, "Design and Creation of a Large-Scale Database of Structural Annotations," in *Proc. of the 12th International Society of Music Information Retrieval*, Miami, FL, USA, 2011, pp. 555–560.

[21] Bee Suan Ong and Perfecto Herrera, "Semantic Segmentation of Music Audio Contents," in *Proc. of the International Computer Music Conference*, Barcelona, Spain, 2005.

[22] Hanna Lukashevich, "Towards Quantitative Measures of Evaluating Song Segmentation," in *Proc. of the 10th International Society of Music Information Retrieval*, Philadelphia, PA, USA, 2008, pp. 375–380.

[23] Meinard Müller and Michael Clausen, "Transposition-Invariant Self-Similarity Matrices," in *Proc. of the 8th International Conference on Music Perception and Cognition*, Vienna, Austria, 2007, pp. 47–50.