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 vari-
Correlation found that the similarities instead of binary values. In our experiments we more generic recurrence plots [16], but using distances (or other distances, including the Euclidean, Cosine or Manhattan distance) gave better results 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 entire matrix between 0 (very dissimilar) and 1 (equal). We illustrate this enhancement in Figure 1.

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.

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 \( X = (x_1, \ldots, x_N) \), which has N row observations \( x_i \) 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 \( z \) a row vector and by \( z^T \) a column one.

C-NMF adds a constrain to \( F = (f_1^T, \ldots, f_r^T) \) such that its columns \( f_j^T \) become convex combinations of the features of \( X \):

\[
f_j^T = x_1 w_{1j} + \ldots + x_p w_{pj} = X w_j^T \quad j \in [1: r] \tag{2}
\]

For a linear combination to be convex, all coefficients \( w_{ij} \) must be positive and the sum of each set of coefficients \( w_{ij} \) must be 1. Formally: \( w_{ij} \geq 0, \sum_j w_{ij} = 1 \).

This results in \( F = X W \), where \( W \in \mathbb{R}^{p \times r} \), which makes the columns \( f_j^T \) 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: \( R_j = f_j^T G_j \), where \( j \in [1: r] \). Finally, C-NMF can be formally characterized as: \( X \approx X W G \).

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 \( x_i \), the similarity of the time frame \( i \) with the rest of the time frames. Thus, each row \( f_i \) 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
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 $M^n = \{R^n_1, \ldots, R^n_r\}$ (where $n$ is the execution index, $n \in [1 : T]$). This is formally illustrated in Equation 3.

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

(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 corrected by the Tampere University of Technology (TUT Beatles)\(^2\). 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
C-NMF outperforms both NMF and SI-PLCA. We can see that $S_o$ is slightly better for SI-PLCA, suggesting that both NMF and C-NMF under-segment the data more than SI-PLCA. However, $S_u$ 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, as described in subsection 3.3.1.

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.

| 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). |
|---|---|---|---|---|---|---|---|---|---|
| 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 | 39.2 | 30.0 | 62.2 |

\[^4\]http://marl.smusic.nyu.edu/resources/siplca-segmentation
6. REFERENCES


