Nonnegative Tucker Decomposition with Beta-divergence for Music Structure Analysis of audio signals

Axel Marmoret[1], Florian Voorwinden[1], Valentin Leplat[2], Jérémy E. Cohen[3], Frederic Bimbot[1]

[1]Univ. Rennes 1, Inria, CNRS, IRISA, France.

[2]Center for Artificial Intelligence Technology (CAIT), Skoltech, Moscow, Russia.

[3]Univ Lyon, INSA-Lyon, UCBL, UJM-Saint Etienne, CNRS, Inserm, CREATIS UMR 5220, Villeurbanne, France
axel.marmoret@irisa.fr, jeremy.cohen@cnrs.fr

Résumé – La décomposition en Tucker nonnégatif (NTD), un modèle de décomposition tensorielle, permet d'extraire des motifs pertinents de manière non-supervisée, notamment dans le cadre de signaux audios. Néanmoins, les algorithmes actuels pour calculer la NTD sont souvent conçus pour optimiser la norme euclidienne. Ce travail propose un algorithme à mises à jour multiplicatives pour optimiser la NTD par rapport à la β-divergence, souvent considérée comme plus pertinente pour traiter des signaux audios. En particulier, cet article montre comment implémenter ces mises à jour de manière efficace en utilisant l'algèbre tensoriel. Finalement, des résultats expérimentaux sur la tâche d'estimation de la structure musicale montrent que la NTD optimisée par rapport à la β-divergence améliore les précédents résultats obtenus par rapport à la norme euclidienne.

Abstract – Nonnegative Tucker decomposition (NTD), a tensor decomposition model, has received increased interest in the recent years because of its ability to blindly extract meaningful patterns, in particular in music information retrieval. Nevertheless, existing algorithms to compute NTD are mostly designed for the Euclidean loss. This work proposes a multiplicative updates algorithm to compute NTD with the β-divergence loss, often considered a better loss for audio processing. We notably show how to implement efficiently the multiplicative rules using tensor algebra. Finally, we show on a music structure analysis task that unsupervised NTD fitted with β-divergence loss outperforms earlier results obtained with the Euclidean loss.

1 Introduction

Tensor factorization models are powerful tools to interpret multi-way data, and are nowadays used in numerous applications [1]. These models allow to extract interpretable information from the input data, generally in an unsupervised (or weakly-supervised) fashion, which can be a great asset when training data is scarcely available. This is the case for music structure analysis (MSA) which consists in segmenting music recordings from the audio signal. For such applications, annotations can be ambiguous and difficult to collect [2].

Nonnegative Tucker decomposition (NTD) has previously proven to be a powerful tool for MSA [3, 4]. While usually the Euclidean distance is used to fit the NTD, audio spectra exhibit large dynamics with respect to frequencies, which leads to a preponderance of few and typically low frequencies when using Euclidean distance. Contrarily, β-divergences, and more particularly Kullback-Leibler and Itakura-Saito divergences, are known to be better suited for time-frequency features. We introduce a new algorithm for NTD where the objective cost is the minimization of the β-divergence, and we study the resulting decompositions as regards their benefit on the MSA task on the RWC-Pop database [5]. The proposed algorithm adapts the multiplicative updates framework well-known for nonnegative matrix factorization (NMF) [6, 7] to the tensor case, detailing efficient tensor contractions. It is closely related to [8], but studies instead the β-divergence and proposes modified multiplicative updates that guarantees global convergence to a stationary point. Code is fully open-source in *nn_fac*[1].

2 Mathematical background

2.1 Nonnegative Tucker Decomposition

NTD is a mathematical model where a nonnegative tensor is approximated as the product of factors (one for each mode of the tensor) and a small core tensor linking these factors. NTD is often used as a dimensionality reduction technique, but it may also be seen as a part-based representation similar to NMF. In this work, we focus on third-order tensors for simplicity. Denoting $\mathcal{X} \in \mathbb{R}_+^{J \times K \times L}$ the tensor to approximate and using conventional tensor-product notation [1], computing the NTD boils down to

[1] https ://gitlab.inria.fr/amarmore/nonnegative-factorization

seek for three nonnegative matrices $W \in \mathbb{R}_+^{J\times J'}$, $H \in \mathbb{R}_+^{K\times K'}$ and $Q \in \mathbb{R}_+^{L\times L'}$ and a core tensor $\mathcal{G} \in \mathbb{R}_+^{J'\times K'\times L'}$ such that :

$$\mathcal{X} \approx \mathcal{G} \times_1 W \times_2 H \times_3 Q \tag{1}$$

This decomposition is also presented in Figure 1.

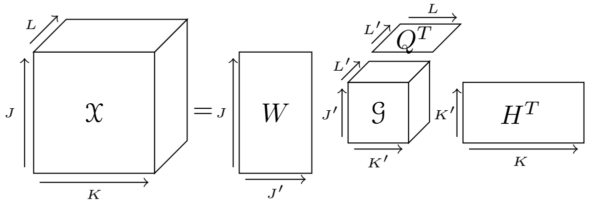

FIGURE 1 – Nonnegative Tucker decomposition of tensor $\mathcal{X}$ in factor matrices W, H, Q, and core tensor $\mathcal{G}$.

NTD is generally performed by minimizing some distance or divergence function between the original tensor and the approximation. Many algorithms found in the literature [3, 9, 10] are based on the minimization of the squared Euclidean distance. In this work, we instead consider the β-divergence, detailed hereafter.

2.2 The β-divergence loss function

In this work, we will focus on the β-divergence function introduced in [11]. Given two nonnegative scalars x and y, the β-divergence between x and y denoted $d_\beta(x|y)$ is defined as follows :

$$d_\beta(x|y) = \begin{cases} \frac{x}{y} - \log(\frac{x}{y}) - 1 & \beta = \; 0 \\ x\log(\frac{x}{y}) + (y-x) & \beta = \; 1 \\ \frac{x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}}{\beta(\beta-1)} & \beta \in \; \mathbb{R}\backslash\{0,1\} \end{cases} \tag{2}$$

This divergence generalizes the Euclidean distance ($\beta = 2$), and the Kullback-Leibler (KL) ($\beta = 1$) and Itakura-Saito (IS) ($\beta = 0$) divergences. The β-divergence $d_\beta(x|y)$ is homogeneous of degree β, that is for any $\lambda \in \mathbb{R}$, we have $d_\beta(\lambda x|\lambda y) = \lambda^\beta d_\beta(x|y)$. It implies that factorizations obtained with $\beta > 0$ (such as the Euclidean distance or the KL divergence) will rely more heavily on the largest data values and less precision is to be expected in the estimation of the low-power components. The IS divergence ($\beta = 0$) is scale-invariant and is the only one in the β-divergences family to possess this property. It implies that entries of low power are as important in the divergence computation as the areas of high power. This property is interesting when processing audio signals as low-power frequency bands can contribute as much as high-power frequency bands to their characterization. Both KL and IS divergences are notoriously known to be better suited to audio source separation than the Euclidean distance [6, 12].

Hence, this work focuses on how to compute a candidate solution to approximate NTD with β-divergence as a loss function :

$$\underset{W\geq 0, H\geq 0, Q\geq 0, \mathcal{G}\geq 0}{\arg\min} \quad D_\beta(\mathcal{X}|\mathcal{G} \times_1 W \times_2 H \times_3 Q) \tag{3}$$

with D_β the elementwise β-divergence.

3 A multiplicative updates algorithm

3.1 Multiplicative updates rules

The cost function is non-convex with respect to all factors, and computing a global solution to NTD is NP-Hard since NTD is a generalization of NMF. However, each subproblem obtained when fixing all but one mode is convex as long as $\beta \in [1, 2]$. Hence, block-coordinate algorithms, that update one factor at a time while fixing all the other factors, are standard to solve both NMF and NTD [7, 10, 12]. In particular, the seminal paper by Lee and Seung [7] proposed an alternating algorithm for NMF with β-divergence, later revisited by Fevotte *et. al.* [12], which we shall extend to NTD.

The multiplicative updates (MU) rule in approximate NMF $M \approx UV^\intercal$ are

$$U \leftarrow \max\left(U \cdot \left(\frac{\left[(UV)^{\cdot(\beta-2)} \cdot M\right] V^\intercal}{(UV)^{\cdot(\beta-1)} V^\intercal}\right)^{\cdot\gamma(\beta)}, \epsilon\right) \tag{4}$$

with $\cdot$ and $\div$ the element-wise product and division, $\epsilon > 0$ a small constant and $\gamma(\beta)$ a function equal to $\frac{1}{2-\beta}$ if $\beta < 1$, 1 if $1 \leq \beta \leq 2$, and $\frac{1}{\beta-1}$ if $2 < \beta$ [12]. The element-wise maximum between the matrix update, *i.e.* the closed form expression of the minimizer of the majorization built at the current iterate, and ϵ in (4) aims at avoiding zero entries in factors, which may cause division by zero, and establishing convergence guarantee to stationary points within the BSUM framework [13].

3.2 Multiplicative updates for NTD

The NTD model can be rewritten using tensor matricization, *e.g.* along the first mode :

$$\begin{aligned} \mathcal{X} &= \mathcal{G} \times_1 W \times_2 H \times_3 Q \\ \Leftrightarrow \mathcal{X}_{(1)} &= W \mathcal{G}_{(1)} (H \otimes Q)^\intercal \end{aligned} \tag{5}$$

where $\mathcal{X}_{(i)}$ is the matricization of the tensor $\mathcal{X}$ along mode i [1] and $\otimes$ denotes the Kronecker product. The matricization are analogous for factors H and Q. One can therefore interpret equation (5) as a NMF of $\mathcal{X}_{(1)}$ with respect to W and $\mathcal{G}_{(1)} (H \otimes Q)^\intercal$.

A difficulty is that forming the Kronecker products is bound to be extremely inefficient both in terms of memory allocation and computation time. Instead, for the MU rules of factor matrices W, H, Q, matrix $V := \mathcal{G}_{(i)} (H \otimes Q)^\intercal$ can be computed efficiently using the identity :

$$\mathcal{G}_{(1)} (H \otimes Q)^\intercal = (\mathcal{G} \times_2 H \times_3 Q)_{(1)} \tag{6}$$

which brings down the complexity of forming V from [2] $\mathcal{O}(KLJ'K'L')$ if done naively to $\mathcal{O}(KJ'K'L' + LJ'KL')$ and drastically reduces memory requirements.

[2] The multiway products are computed in lexicographic order.

For the core factor, one can use the vectorization property

$$\mathrm{vec}(\mathcal{X}) = (W \otimes H \otimes Q)\mathrm{vec}(\mathcal{G}) \ , \tag{7}$$

to relate the core update with the NMF MU rules. Again matrix $U := W \otimes H \otimes Q$ is $J'K'L'$ times larger than the data itself. Therefore, for any vector $t := \mathrm{vec}(\mathcal{T})$ we use the identity

$$(W \otimes H \otimes Q)\, t = \mathrm{vec}(\mathcal{T} \times_1 W \times_2 H \times_3 Q) \ . \tag{8}$$

Products $U^\intercal t$ are computed similarly.

Algorithm 1 shows one loop of the proposed MU algorithm. The overall complexity of such an iteration is dependant on the multiway product effective complexity, but is no worse than $\mathcal{O}(JKLJ')$ if $J, K, L > J' > K', L'$. The proposed MU rules do not increase the cost at each iteration and for any initial factors, every limit point is a stationary point [14, Theorem 8.9].

```
Algorithm 1: A loop of $\beta$_NTD($\mathcal{X}$,dimensions,$\beta$)
  Input: $\mathcal{X}, \mathcal{G}, W, H, Q, \epsilon, \beta$
  Output: $\mathcal{G}, W, H, Q$
  $V = (\mathcal{G} \times_2 H \times_3 Q)_{(1)}$
  $W \leftarrow \max\left(W \cdot \left(\frac{[(WV)^{\cdot(\beta-2)} \cdot \mathcal{X}_{(1)}]V^\intercal}{(WV)^{\cdot(\beta-1)}V^\intercal}\right)^{\cdot\gamma(\beta)}, \epsilon\right)$
  Perform analogous updates for $H$ and $Q$
  $\mathcal{N} = (\mathcal{G} \times_1 W \times_2 H \times_3 Q)^{\cdot(\beta-2)} \cdot \mathcal{X}$
  $\mathcal{D} = (\mathcal{G} \times_1 W \times_2 H \times_3 Q)^{\cdot(\beta-1)}$
  $\mathcal{G} \leftarrow \max\left(\mathcal{G} \cdot \left(\frac{\mathcal{N} \times_1 W^\intercal \times_2 H^\intercal \times_3 Q^\intercal}{\mathcal{D} \times_1 W^\intercal \times_2 H^\intercal \times_3 Q^\intercal}\right)^{\cdot\gamma(\beta)}, \epsilon\right)$
```

4 Experimental Framework

4.1 NTD for music processing

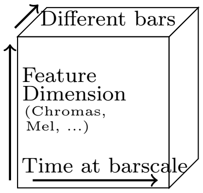

FIGURE 2 – TFB tensor

NTD has already been introduced to process audio signals, and was shown to provide a barwise pattern representation of a music [3, 4]. For music, NTD is performed on a 3rd-order tensor, called TFB tensor, which is the result of splitting a spectrogram on bar frontiers and concatenating the subsequent barwise spectrograms on a 3rd-mode. Hence, a TFB tensor is composed of a frequential mode and two time-related modes : an inner-bar (low-level) time, and a bar (high-level) time. Each bar contains 96 frames, which are selected as equally spaced on a oversampled spectrogram (hop length of 32 samples), in order to account for bar length discrepancies [3].

In our previous work [3], we computed the NTD on chromagrams, then evaluated on music segmentation. We extend this previous study to Mel-spectrograms, which consist in the STFT of a song with frequencies aggregated following a Mel filter-bank. They provide a richer representation of music than chromagrams but they live in a higher dimensional space. On the basis of this alternate representation, we compare the algorithm introduced in [3] (HALS-based NTD with Euclidean loss minimization, see [10]) with the proposed algorithm in the MSA task, on the audio signals of the RWC Pop database [5], which is a standard dataset in music information retrieval.

In practice, Mel-spectrograms are dimensioned following the work of [15], which is considered as state-of-the-art in this task. Precisely, STFT are computed as power spectrograms with a window size of 2048 samples for a signal sampling rate of 44.1 kHz. A Mel-filter bank of 80 triangular filter between 80 Hz and 16 kHz is then applied. In addition to this raw Mel representation, we study a logarithmic variant, which is generally used as a way to account for the exponential distribution of power in audio spectra. As the logarithmic function is negative for values lower than 1, we introduce the Nonnegative Log-Mel spectrogram (NNLMS) as $\mathrm{NNLMS} = \log(\mathrm{Mel} + 1)$.

Finally, each nonnegative TFB tensor has sizes $80 \times 96 \times L$, with L the number of bars in the song. When (empirically) setting core dimensions $J', K', L' = 32$, one iteration of the MU algorithm takes approximately 0.2s, while one iteration of the HALS algorithm takes approximately 0.75s on an Intel®Core™i7 processor. Nonetheless, the HALS algorithm generally converges in less iterations than MU.

4.2 Music structure analysis based on NTD

MSA consists in segmenting a song into sections (such as "verse", "chorus", etc) as presented in [2]. The goal here is to retrieve the boundaries between different sections. We use the same segmentation framework than in [3].

Segmentation results are presented in Table 1, where we compare the performance of segmenting the chromagram using the HALS-NTD [3] with segmenting Mel and NNLMS representations. These results show that, for both representations, using the KL and IS divergences instead of the Euclidean loss enhance segmentation performance. Segmentation results are also higher when using NTD on the NNLMS rather than on the Mel-spectrogram, and outreach previous results on Chromagrams. Hence, adapting the decomposition to the dynamics of audio signals seems beneficial, both in term of loss function and feature.

4.3 Qualitative assessment of patterns

As a qualitative study between the different β-divergences ($\beta \in \{0, 1, 2\}$), we computed the NTD with these three values on the STFT of the song "Come Together" by the Beatles. Using the Griffin-Lim algorithm [16] and softmasking [4], spectrograms computed with the NTD (such as musical patterns $W\mathcal{G}_{[:,:,i]}H^T$ [3]) are reconstructed into listenable signals. Results are available online [3], and qua-

[3] https ://ax-le.github.io/resources/examples/ListeningNTD.html

Table 1 – Segmentation results on the RWC Pop dataset [5], with different loss functions. P, R and F respectively represent Precision, Recall and F-measure, based on the evaluation of correct and incorrect boundaries (in time). These metrics are computed with two tolerances for considering a boundary correct : 0.5s and 3s. These values are standard in MSA [2]. Core dimensions J', K', L' are fitted among values $\{8, 16, 24, 32, 40\}$ with two-fold cross-validation, by splitting the RWC-Pop dataset between even and odd songs.

Representation	Technique		$P_{0.5}$	$R_{0.5}$	$F_{0.5}$	P_3	R_3	F_3
Chromas (initial work [3])	HALS-NTD		55.3%	59.3%	56.6%	70.3%	75.1%	71.9%
Mel-spectrogram	HALS-NTD ($\beta = 2$)		45.9%	49.5%	47.2%	68.1%	73.0%	69.7%
	MU-NTD	$\beta = 1$	53.3%	56.9%	54.6%	70.5%	75.3%	72.2%
		$\beta = 0$	51.1%	56.8%	53.3%	69.9%	78.0%	73.1%
NNLMS	HALS-NTD ($\beta = 2$)		50.5%	52.7%	51.1%	71.2%	74.6%	72.2%
	MU-NTD	$\beta = 1$	**57.8%**	**61.9%**	**59.3%**	73.9%	79.3%	75.9%
		$\beta = 0$	55.9%	61.7%	58.1%	**74.0%**	**81.9%**	**77.1%**

litatively confirm that the KL divergence is better adapted to signals than the Euclidean loss, while this is more contrasted for the IS divergence.

5 Conclusion

Nonnegative Tucker decomposition is able to extract salient patterns in numerical data. This article has proposed a tractable and globally convergent algorithm to perform the NTD with the β-divergence as loss function. This appears to be of particular interest for a wide range of signals and applications, notably in the audio domain, as supported in this paper by quantitative results on a music structure analysis task and qualitative examples for $\beta = 1, 0$.

Future work may consider the introduction of sparsity constraints, which generally improve the interpretability of nonnegative decompositions, and seeking additional strategies to accelerate the algorithm itself.

Références

[1] T.G. Kolda and B.W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, September 2009.

[2] O. Nieto et al., "Audio-based music structure analysis : Current trends, open challenges, and applications," *Trans. of the Int. Soc. for Music Information Retrieval*, vol. 3, no. 1, 2020.

[3] A. Marmoret, J.E. Cohen, N. Bertin, and F. Bimbot, "Uncovering audio patterns in music with nonnegative Tucker decomposition for structural segmentation," in *ISMIR*, 2020, pp. 788–794.

[4] J.B.L. Smith and M. Goto, "Nonnegative tensor factorization for source separation of loops in audio," in *IEEE ICASSP*. IEEE, 2018, pp. 171–175.

[5] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database : Popular, Classical and Jazz Music Databases," in *ISMIR*, 2002, vol. 2, pp. 287–288.

[6] C. Févotte, N. Bertin, and J.L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence : With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[7] D.D. Lee and H.S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[8] Yong-Deok Kim, Andrzej Cichocki, and Seungjin Choi, "Nonnegative tucker decomposition with alpha-divergence," in *IEEE ICASSP*. IEEE, 2008, pp. 1829–1832.

[9] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, "Efficient nonnegative Tucker decompositions : Algorithms and uniqueness," *IEEE Trans. on Image Processing*, vol. 24, no. 12, pp. 4990–5003, 2015.

[10] A.H. Phan and A. Cichocki, "Extended HALS algorithm for nonnegative Tucker decomposition and its applications for multiway analysis and classification," *Neurocomputing*, vol. 74, no. 11, pp. 1956–1969, 2011.

[11] A. Basu, I.R. Harris, L.N. Hjort, and M.C. Jones, "Robust and efficient estimation by minimising a density power divergence," *Biometrika*, vol. 85, no. 3, pp. 549–559, 1998.

[12] C. Févotte and J. Idier, "Algorithms for nonnegative matrix factorization with the β-divergence," *Neural computation*, vol. 23, no. 9, pp. 2421–2456, 2011.

[13] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[14] N. Gillis, *Nonnegative Matrix Factorization*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2020.

[15] T. Grill and J. Schlüter, "Music boundary detection using neural networks on combined features and two-level annotations," in *ISMIR*, 2015, pp. 531–537.

[16] D. Griffin and J. Lim, "Signal estimation from modified short-time fourier transform," *IEEE Trans. on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.