Bayesian non-negative matrix factorization with learned temporal smoothness priors

Mathieu Coïc and Juan José Burred

Audionamix
114, avenue de Flandre
75019 Paris, France
{firstname.{middlename.}lastname}@audionamix.com

Abstract. We combine the use of a Bayesian NMF framework to add temporal smoothness priors, with a supervised prior learning of the smoothness parameters on a database of solo musical instruments. The goal is to separate main instruments from realistic mono musical mixtures. The proposed learning step allows a better initialization of the spectral dictionaries and of the smoothness parameters. This approach is shown to outperform the separation results compared to the unsupervised version.

1 Introduction

Non-negative matrix factorization (NMF) is a well-known signal decomposition technique frequently used for sound source separation. NMF decomposes a spectrogram into a set of spectral bases, each one multiplied by a time-varying weight. When dealing with musical mixtures, it is possible to exploit the specific properties of musical instruments, such as the typical temporal evolution of their spectral bases.

One way of integrating such a priori information is by using statistical priors in a Bayesian statistical framework. This was the approach used to force temporal smoothness in [1], and both temporal smoothness and harmonicity in [2]. Another option is to use supervised methods and perform a prior learning based on a database of isolated instrumental sounds. An example of this second approach is the work presented in [3], where NMF is combined with a pre-trained Hidden Markov Model (HMM) to model dynamic behavior.

In this contribution, we use a combination of both Bayesian priors and database learning to model temporal smoothness and improve separation quality. The goal is to extract the lead instrument from realistic mono musical mixtures. In particular, our system is based on a Bayesian NMF model with temporal smoothness priors described by Inverse Gamma (IG) distributions (Sect. 2), as was done in [1, 2]. Here, we extend such approach by introducing a learning stage, which is based on performing NMF optimization on isolated instruments with the IG parameters as additional optimization parameters (Sect. 3). We evaluate the performance with 4 different instruments, and for all settings (with

or without priors, with or without learning), we compare the performance of two possible implementations of NMF optimization, one based on Multiplicative Updates (NMF-MU), and one based on Expectation-Maximization (NMF-EM).

2 Unsupervised Algorithms

2.1 NMF Framework

The input signal is first transformed into the time-frequency domain by means of a Short Time Fourier Transform (STFT), yielding a matrix $\mathbf{X}$. As in [1], the squared modulus of each element is computed to obtain a matrix of power spectral densities $\mathbf{V} = |\mathbf{X}|^{\circ 2}$. The goal of NMF is to find the non-negative matrices $\mathbf{W}$ and $\mathbf{H}$ such that

$$\mathbf{V} \approx \mathbf{WH}. \tag{1}$$

$\mathbf{W}$ and $\mathbf{H}$ have dimensions $F \times K$ and $K \times N$, respectively, and it is desirable that $F \times K + K \times N \ll FN$. The rows of $\mathbf{H}$ are usually called *activations* and the columns of $\mathbf{W}$ *atoms* or *bases*.

Such factorization is formulated here as the minimization problem

$$\{\mathbf{W}, \mathbf{H}\} = \underset{\mathbf{W},\mathbf{H}\geq 0}{\operatorname{argmin}}\, D_{IS}(\mathbf{V}|\mathbf{WH}), \tag{2}$$

where D_{IS} is a matrix cost function involving the Itakura-Saito element-wise divergence d_{IS}:

$$D_{IS}(\mathbf{V}|\mathbf{WH}) = \sum_{f=1}^{F}\sum_{n=1}^{N} d_{IS}(\mathbf{V}_{(f,n)}|[\mathbf{WH}]_{(f,n)}). \tag{3}$$

The IS divergence, defined as

$$d_{IS}(x \mid y) = \frac{x}{y} - \log\frac{x}{y} - 1, \tag{4}$$

is a good measure for the perceptual difference between two spectra, which is explained by its scale invariance: $d_{IS}(\gamma x|\gamma y) = d_{IS}(x|y)$, for a given scalar γ.

It can be shown [1] that the above optimization (Eq. 2) is equivalent to a Maximum Likelihood (ML) estimation if the columns of the STFT matrix $\mathbf{X}$, denoted by $\mathbf{x}_n$, are supposed to be generated by a K-component Gaussian Mixture Model (GMM):

$$\mathbf{x}_n = \sum_{k=1}^{K} \mathbf{c}_{kn} \in \mathbb{C}^F, \quad \forall n = 1, ..., N, \tag{5}$$

where latent variables $\mathbf{c}_{kn}$ are independent and follow a zero-mean multivariate normal distribution $\mathbf{c}_{kn} \sim \mathcal{N}(0, h_{kn}\mathrm{diag}(\mathbf{w}_k))$, where h_{kn} are the elements of the activation matrix $\mathbf{H}$ and $\mathbf{w}_k$ are the columns of the dictionary matrix $\mathbf{W}$. The

separation process consists in optimizing the criterion $C_{ML}(\boldsymbol{\theta}) \stackrel{\triangle}{=} \log p(\mathbf{V} \mid \boldsymbol{\theta})$, where $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{H}\}$ is the parameter vector.

We implement and test two NMF algorithms, one based on Multiplicative Update rules (NMF-MU), and one based on an EM algorithm (NMF-EM). They mainly differ in their speed of convergence to a global solution and in computational performance. The first one was used in [1], and the second one in [2], and both can be adapted to a Bayesian setting.

2.2 NMF-MU algorithm

Multiplicative Update (MU) rules to iteratively find the optimal $\mathbf{W}$ and $\mathbf{H}$ are given for the IS divergence [4] by

$$\mathbf{H} \leftarrow \mathbf{H} \circ \frac{\mathbf{W}^T \left((\mathbf{W}\mathbf{H})^{\circ[-2]} \circ \mathbf{V} \right)}{\mathbf{W}^T (\mathbf{W}\mathbf{H})^{\circ[-1]}}, \tag{6}$$

$$\mathbf{W} \leftarrow \mathbf{W} \circ \frac{\left((\mathbf{W}\mathbf{H})^{\circ[-2]} \circ \mathbf{V} \right) \circ \mathbf{H}^T}{(\mathbf{W}\mathbf{H})^{\circ[-1]} \mathbf{H}^T}, \tag{7}$$

where the $\circ$ symbol denotes element-wise operations, and the division is also element-wise.

2.3 NMF-EM Algorithm

An alternative to MU is to directly perform an ML estimation of the generative model of Eq. 5 via an EM algorithm. In particular, the Space Alternating Generalized EM (SAGE) algorithm [1] is a type of EM algorithm that allows to update large parameter matrices in separate chunks, with fast convergence properties. In particular, we aim at estimating separately the parameters $\mathbf{C}_k = (\mathbf{c}_{k1}, ..., \mathbf{c}_{kN})$. If we partition the parameter space by $\boldsymbol{\theta} = \bigcup_{k=1}^{K} \boldsymbol{\theta}_k$ where $\boldsymbol{\theta}_k = \{\mathbf{w}_k, \mathbf{h}_k\}$, SAGE consists in choosing for each subset $\boldsymbol{\theta}_k$ a *hidden-data space* which is complete for this particular subset, i.e. $\boldsymbol{\theta}_k = \mathbf{C}_k$. The resulting algorithm to estimate $\mathbf{W}$ and $\mathbf{H}$ is defined in detail in [1].

2.4 Bayesian NMF with temporal smoothness prior

The Bayes rule allows to switch from a ML estimation to a Maximum A Posteriori (MAP) estimation. We can thus introduce the prior distributions $p(\mathbf{W})$ and $p(\mathbf{H})$ in this manner:

$$p(\mathbf{W}, \mathbf{H} \mid \mathbf{V}) = \frac{p(\mathbf{V} \mid \mathbf{W}, \mathbf{H})\, p(\mathbf{W}) p(\mathbf{H})}{p(\mathbf{V})}. \tag{8}$$

In the case of temporal modeling, $p(\mathbf{H})$ is the relevant prior. MAP estimation is obtained by maximizing the following criterion:

$$C_{MAP}(\boldsymbol{\theta}) \stackrel{\triangle}{=} \log p(\boldsymbol{\theta} \mid \mathbf{V}) \stackrel{c}{=} C_{ML}(\boldsymbol{\theta}) + \log p(\mathbf{H}), \tag{9}$$

where the binary operator $\stackrel{c}{=}$ denotes equality up to an additive constant.

Based on the MAP estimator, [1] and [2] propose a Markov chain prior structure to model $p(\mathbf{H})$:

$$p(h_k) = p(h_{k1}) \prod_{n=2}^{N} p(h_{kn} \mid h_{k,n-1}). \tag{10}$$

The main objective is to assure smoothness over the rows of $\mathbf{H}$. With an appropriate choice of the Markov transition matrix, we can favor a slow variation of $\mathbf{h}_k$. For example, we can force $p\left(h_{kn} \mid h_{k,n-1}\right)$ reach its maximum at $p(h_{k,n-1})$. The authors propose:

$$p\left(h_{kn} \mid h_{k,n-1}\right) = \mathcal{IG}\left(h_{kn} \mid \alpha_k, (\alpha_k + 1)h_{k,n-1}\right), \tag{11}$$

where $\mathcal{IG}(x \mid \alpha, \beta)$ is the inverse-Gamma distribution[1] with mode $\frac{\beta}{\alpha+1}$ and the initial distribution $p(h_{k1})$ is Jeffrey's non-informative prior: $p(h_{k1}) \propto \frac{1}{h_{k1}}$. Hence, α_k is a parameter that controls the degree of smoothness for the k-th component.

Note that we can have different smoothness parameters for each component. Thus, the smoothness parameter is actually a vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K)$. In practice, we want to set a smoothness prior only to those components that are supposed to describe the lead instrument. If we assign the first K_s components to the lead instrument, and the remaining ones to the accompaniment, then the α_k priors apply only to $1 \leq k \leq K_s$, and no priors are used for $K_s < k \leq K$.

The priors can be added to both NMF-MU [2] and NMF-EM [1] algorithms, as follows:

– **NMF-MU/IG algorithm**. Eq. (9) gives the following new update rules for $\mathbf{H}$, that replace Eq. 6:

$$h_{k1} \leftarrow h_{k1} \times \left(\frac{\sum_{f=1}^{F} \frac{v_{f1} w_{fk}}{\hat{v}_{f1}^2} + \frac{\alpha_k + 1}{h_{k1}}}{\sum_{f=1}^{F} \frac{w_{fk}}{\hat{v}_{f1}} + \frac{\alpha_k + 1}{h_{k2}}} \right)^{\eta} \tag{12}$$

$$h_{kn} \leftarrow h_{kn} \times \left(\frac{\sum_{f=1}^{F} \frac{v_{fn} w_{fk}}{\hat{v}_{fn}^2} + \frac{(\alpha_k + 1)h_{n-1}}{h_{kn}^2}}{\sum_{f=1}^{F} \frac{w_{fk}}{\hat{v}_{fn}} + \frac{1}{h_{kn}} + \frac{\alpha_k + 1}{h_{k,n+1}}} \right)^{\eta} \tag{13}$$

$$h_{kN} \leftarrow h_{kN} \times \left(\frac{\sum_{f=1}^{F} \frac{v_{fN} w_{fk}}{\hat{v}_{fN}^2} + \frac{(\alpha_k + 1)h_{N-1}}{h_{kN}^2}}{\sum_{f=1}^{F} \frac{w_{fk}}{\hat{v}_{fN}} + \frac{\alpha_k + 1}{h_{kN}}} \right)^{\eta}, \tag{14}$$

where $\eta \in]0, 1]$ plays the role of the step size in gradient descent.

[1] $\mathcal{IG}\left(x \mid \alpha, \beta\right) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} x^{-(\alpha+1)} \exp\left(-\frac{\beta}{x}\right)$

	p_2	p_1	p_0
h_{k1}	$\frac{\alpha_k+1}{\hat{h}_{k2}}$	$F-\alpha_k+1$	$-F\hat{h}_{k1}$
h_{kn}	$\frac{\alpha_k+1}{\hat{h}_{kn+1}}$	$F+1$	$-F\hat{h}_{kn}-(\alpha_k+1)\,h_{k,n-1}$
h_{kN}	0	$F+\alpha_k+1$	$-F\hat{h}_{kN}-(\alpha_k+1)\,h_{k,N-1}$

Table 1. Coefficients for the post estimation of h_{kn} in NMF-EM/IG.

– **NMF-EM/IG algorithm**. To integrate the temporal smoothness prior into NMF-EM, the best way is to add a post estimation after each update, computed as follows:

$$h_{kn} = \frac{\sqrt{p_1^2 - 4p_2p_0} - p_1}{2p_2}, \tag{15}$$

where the coefficients p_0, p_1 and p_2 depend on n and are given in Table 1. In a more recent work [5], a simpler procedure, leading to a better-posed optimization problem and based on Majorization-Minimization (MM), has been proposed as an alternative to a Bayesian EM approach as described above. In the present paper, we use EM as proposed in [1], and will explore the MM alternative in the future.

3 Supervised Algorithms

The smoothness priors α_k defined in the previous section need to be set by hand prior to separation, and remain fixed throughout the optimization process. Furthermore, it would be too cumbersome to find good manual parameters for the individual priors of components with indices $1 \leq k \leq K_s$. Thus, an improvement of separation quality is expected if the α_ks are automatically learned from a training database of isolated instrumental excerpts.

We implement learning by considering the smoothness vector $\boldsymbol{\alpha}$ as an additional parameter to optimize, obtaining the new parameter vector

$$\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{H}, \boldsymbol{\alpha}\}. \tag{16}$$

A MAP estimation (Eq. 9) is performed on an audio file containing concatenated solo excerpts. We keep the estimated dictionary matrix $\hat{\mathbf{W}}$ and the smoothness vector $\hat{\boldsymbol{\alpha}}$ obtained in this way, and use them to initialize the MAP estimation performed on the mixture for actual separation.

The new update rule for the α_k coefficients is derived via ML estimation given the IG Markov chain from Eqs. 10 and 11. The log-likelihood is given by

$$\begin{aligned}\log\left(p(h_k)\right) &\stackrel{c}{=} \log\left(\frac{1}{h_{k1}}\right) + \sum_{n=2}^{N}\Big[\alpha_k \log((\alpha_k+1)h_{k,n-1}) - \log(\Gamma(\alpha_k)) \qquad (17)\\ &\quad -(\alpha_k+1)\log(h_{kn}) - \frac{\alpha_k h_{k,n-1}}{h_{kn}} - \frac{h_{k,n-1}}{h_{kn}}\Big]\\ &\stackrel{c}{=} \alpha_k(\log(h_{k1}) - \log(h_{kN})) - \log(h_{k1}) + \sum_{n=2}^{N}\left[\alpha_k\log(\alpha_k+1)\right. \qquad (18)\\ &\quad \left. - \log(\Gamma(\alpha_k)) - \log(h_{kn}) - \frac{\alpha_k h_{k,n-1}}{h_{kn}} - \frac{h_{k,n-1}}{h_{kn}}\right].\end{aligned}$$

Minimizing the ML criterion gives:

$$\begin{aligned}&\frac{\partial \log(p(h_k))}{\partial \alpha_k} = 0\\ &\Leftrightarrow\ \log\left(\frac{h_{k1}}{h_{kN}}\right) + \sum_{n=2}^{N}\left[\log(\alpha_k+1) + \frac{\alpha_k}{\alpha_k+1} - \psi(\alpha_k) - \frac{h_{k,n-1}}{h_{kn}}\right] = 0\\ &\Leftrightarrow\ \log(\alpha_k+1) + \frac{\alpha_k}{\alpha_k+1} - \psi(\alpha_k) = \frac{1}{N-1}\left(\log\left(\frac{h_{kN}}{h_{k1}}\right) + \sum_{n=2}^{N}\frac{h_{k,n-1}}{h_{kn}}\right),\end{aligned} \qquad (19)$$

where ψ is the digamma function defined as: $\psi(x) = \frac{\Gamma'(x)}{\Gamma(x)}$. Since Eq. 19 has no closed-form solution, the estimation of the current α_k is computed numerically.

For separation, we assign again the first K_s components to the main instrument. Thus, learned vector $\hat{\boldsymbol{\alpha}}$ applies only to the first K_s components of the matrix $\mathbf{H}$ is separation, and the first K_s columns of $\mathbf{W}$ are equal to the learned dictionary $\hat{\mathbf{W}}$.

4 Evaluation

For learning, 4 instruments from the RWC musical instrument sound database [6] were used. The instruments chosen were saxophone, trumpet, classical guitar and piano. The saxophone and the trumpet are melodic instruments which usually play only one note at a time. The piano and the guitar are polyphonic instruments that can play several notes at a time (although the guitar will mostly play individual notes when doing a solo). Furthermore, saxophone and trumpet are sustained instruments (the notes can be held as long as the breathing of the player allows), whereas piano and guitar are non-sustained instruments with note energy always decaying after the onset. Thus, the smoothness parameters over the rows of $\mathbf{H}$ are expected to be quite different between both kinds of instruments.

500 iterations		Unsupervised		Supervised	
		Without priors	IG	Without priors	IG
Saxophone	NMF-MU	10.19	9.12	**10.47**	9.66
	NMF-EM	7.14	6.76	7.01	7.58
Trumpet	NMF-MU	6.16	4.80	6.39	**7.88**
	NMF-EM	3.63	3.98	4.55	5.06
Classic Guitar	NMF-MU	9.84	8.75	8.88	**10.07**
	NMF-EM	7.38	7.52	8.01	6.52
Piano	NMF-MU	**6.99**	4.73	5.44	6.95
	NMF-EM	3.11	3.98	2.97	2.08
Global	NMF-MU	8.30	6.85	7.80	**8.64**
	NMF-EM	5.32	5.56	5.64	5.31

Table 2. Average SDR (in dB)

To evaluate separation, 18 mixes were created from songs available in multitrack and featuring solos by those instruments[2]. For each song, one mono track was created for the solo, and one mono track containing all the remaining instruments (accompaniment).

For objective evaluation in terms of Source to Distortion Ratio (SDR), we use the BSS_EVAL toolbox [7]. After separation into K NMF components, it is still necessary to assign the components to one of the sources. For evaluation purposes, SDR is measured between each component and the original tracks. The higher SDR determines if the component represents the solo or the accompaniment.

We evaluate both NMF-MU and NMF-EM algorithms, with or without priors, and in both unsupervised and supervised versions. In the unsupervised version, the parameters are initialized randomly except for the smoothness parameters, which are fixed empirically. In that case, we set the same smoothness value for all α_k. In the supervised case, the learned parameters are then used in the separation process to initialize the system, as explained in Sect 3. Note that in the supervised version without smoothness priors, the dictionary $\mathbf{W}$ is learned anyway.

Results are given in Table 2. The following conclusions can be drawn:

- NMF-MU algorithms perform in general better than NMF-EM algorithms.
- In unsupervised algorithms, using the IG smoothness priors is not efficient. This is probably due to the difficulty of manually finding good values for $\hat{\boldsymbol{\alpha}}$.
- Supervised algorithms outperform unsupervised algorithms, except in the case of the piano, in which case the maximum performance is virtually the same. This might indicate that the IG distribution is not well suited to describe the dynamics of the piano spectra.
- In supervised algorithms, using the smoothness priors improves performance, except for the saxophone.

[2] Source: `ccmixter.org`

A selection of sound examples can be found online[3].

5 Conclusions and perspectives

We have proposed a learning stage for the IG temporal smoothness priors within an NMF Bayesian framework for separation of main instruments in mono mixtures. An evaluation of the different system configurations was performed, including supervised and unsupervised versions, with or without priors, and both in MU and EM implementations, for sustained (trumpet and saxophone) and non-sustained (piano and guitar) instruments. Supervised approaches are shown to perform better than unsupervised ones, except in the case of the piano. Globally, the MU versions of the algorithms perform better.

A refinement of the temporal priors will be subject to further study. In particular, a temporal description will probably benefit from a structured representation considering the attack and sustain parts separately. Also, other prior distributions will be investigated to improve difficult cases, such as the piano. Finally, other instrument-specific priors, such as spectral smoothness or harmonicity, might also be taken into account in order to further improve separation quality.

References

1. C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis. *Neural Computation*, 21(3):793–830, March 2009.
2. N. Bertin, R. Badeau, and E. Vincent. Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Transactions on Audio, Speech and Language Processing*, 18(3):538–549, March 2010.
3. G. J. Mysore and P. Smaragdis. A non-negative approach to semi-supervised separation of speech from noise with the use of temporal dynamics. In *Proc. IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011.
4. A. Cichocki, R. Zdunek, and S.-I. Amari. Csiszár's divergences for non-negative matrix factorization: Family of new algorithms. In *Independent Component Analysis and Blind Signal Separation, LNCS-3889*, pages 32–39. Springer, 2006.
5. C. Févotte. Majorization-minimization algorithm for smooth Itakura-Saito nonnegative matrix factorization. In *Proc. IEEE Int. Conf. on Audio, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, May 2011.
6. M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Music genre database and musical instrument sound database. *Proc. of the 4th International Conference on Music Information Retrieval (ISMIR)*, pages 229–230, 2003.
7. C. Févotte, R. Gribonval, and E. Vincent. BSS_EVAL toolbox user guide. *IRISA Technical Report 1706*, Rennes, France, 2005.

[3] `http://audionamix.com/BayesianNMF1/`