Non negative sparse representation for Wiener based source separation with a single sensor

Laurent Benaroya, Lorcan Mcdonagh, Frédéric Bimbot, Rémi Gribonval

NON NEGATIVE SPARSE REPRESENTATION FOR WIENER BASED SOURCE SEPARATION WITH A SINGLE SENSOR

Laurent BENAROYA, Lorcan M. DONAGH, Frédéric BIMBOT, Rémi GRIBONVAL

IRISA (CNRS & INRIA), METISS, Campus de Beaulieu
35042 Rennes Cedex, France

ABSTRACT

We propose a new method to perform the separation of two sound sources from a single sensor. This method generalizes the Wiener filtering with locally stationary, non gaussian, parametric source models. The method involves a learning phase for which we propose three different algorithm. In the separation phase, we use a sparse non negative decomposition algorithm of our own. The algorithms are evaluated on the separation of real audio data.

1. INTRODUCTION

We propose a new method to perform the separation of two sound sources from a single sensor. That is to say, we observe $x(t) = s_1(t) + s_2(t)$ and we want to estimate $s_1(t)$ and $s_2(t)$.
If s_1 and s_2 are stationary gaussian, the optimal estimates are given by Wiener filtering, which splits each frequency component of x into a contribution of each source by relying on their respective power spectral densities (PSD) [1]:

$$\mathcal{F}\hat{s}_1(f) = \frac{\sigma_1^2(f)}{\sigma_1^2(f) + \sigma_2^2(f)}\mathcal{F}x(f)$$
$$\mathcal{F}\hat{s}_2(f) = \frac{\sigma_2^2(f)}{\sigma_1^2(f) + \sigma_2^2(f)}\mathcal{F}x(f)$$

where $\mathcal{F}$ is the Fourier transform.
Here, we are interested in the larger class of locally stationary (non gaussian) sources and we try to generalize the Wiener filtering. We naturally work with the short term Fourier transform (STFT) denoted by $\mathcal{S}$.
A simple parametric model of a locally stationary source is

$$s_i(t) = a_i(t) \times b_i(t)$$

where $a_i(t) \geq 0$ is the amplitude parameter and $b_i(t)$ is a stationary gaussian process with PSD $\sigma_i^2(f)$, with $i = 1, 2$. The amplitude parameter is supposed here to be slowly varying compared to the length of the window that is used in the STFT, that is: $\mathcal{S}s_i(f,t) \approx a_i(t) \times \mathcal{S}b_i(f,t)$.

If we are able to estimate $a_1(t)$ and $a_2(t)$, the Wiener filtering becomes [2]

$$\mathcal{S}\hat{s}_1(f,t) = \frac{a_1(t)\sigma_1^2(f)}{a_1(t)\sigma_1^2(f) + a_2(t)\sigma_2^2(f)}\mathcal{S}x(f,t)$$
$$\mathcal{S}\hat{s}_2(f,t) = \frac{a_2(t)\sigma_2^2(f)}{a_1(t)\sigma_1^2(f) + a_2(t)\sigma_2^2(f)}\mathcal{S}x(f,t)$$

For independent sources, we may estimate the parameters $a_i(t)$ using the formula $|\mathcal{S}x(f,t)|^2 \approx a_1(t)\sigma_1^2(f) + a_2(t)\sigma_2^2(f)$. This simple model is a bit crude to describe real audio sources, which may present different timbres or pitches corresponding to different spectral shapes at different times. Therefore, we propose the following generalized model:

$$s_i(t) = \sum_{k \in K_i} a_k(t) b_k(t)$$

where $b_k(t)$ is a stationary gaussian process with spectral shapes corresponding to the PSD $\sigma_k^2(f)$.K_1, K_2 are index sets with $K_1 \cap K_2 = \oslash$. the $a_k(t)$ are slowly varying amplitude parameters.
We summarize below the general framework of our study

General Framework 1

1. Learn the PSD $\sigma_k^2(f)$ on training samples of the sources.
2. Decompose

$$|\mathcal{S}x(f,t)|^2 \approx \sum_{k \in K_1 \cup K_2} a_k(t)\sigma_k^2(f)$$

   Under constraints $\forall k$, $a^k(t) \geq 0$
3. Estimate

$$\widehat{\mathcal{S}s}_1(f,t) = \frac{\sum_{k \in K_1} a_k(t)\sigma_k^2(f)}{\sum_{k \in K_1 \cup K_2} a_k(t)\sigma_k^2(f)}\mathcal{S}x(f,t)$$
$$\widehat{\mathcal{S}s}_2(f,t) = \frac{\sum_{k \in K_2} a_k(t)\sigma_k^2(f)}{\sum_{k \in K_1 \cup K_2} a_k(t)\sigma_k^2(f)}\mathcal{S}x(f,t)$$

Section 2 : we discuss the learning phase for both PSD sets. Three methods will be covered : a plain randomized algorithm, an algorithm based on a correlation function and an algorithm based on additive mixture of PSD.
Section 3 : we present a new algorithm for sparse decomposition with non negative coefficients constraints.
Section 4 : we evaluate the algorithms on real audio data with different size of PSD set, K_1 and K_2.

2. LEARNING THE PSD SETS

Given samples $s(t_1), \ldots, s(t_N)$ of a source s, we aim here at extracting a set of PSD vectors $\{\sigma_k^2\}_{k \in K_i}$, with $\text{Card}(K_i) = m$, representative of the spectra $|\mathcal{S}s(t,f)|^2$ of this source.
$\forall t \; \exists (a_1, \ldots, a_m) \; \forall f \; |\mathcal{S}s(t,f)|^2 \approx \sum_{k \in K_i} a_k(t) \sigma_k^2(f)$.
As the PSD $\sigma_k^2(f)$ are only defined up to multiplicative constant, we will suppose that $\int \sigma_k^2(f) df = 1$.
We use here the notation $g_t(f) = \frac{|\mathcal{S}s(t,f)|^2}{\int |\mathcal{S}s(t,f)|^2 df}$ for the normalized spectral vectors of the training signal $s(t)$.

2.1. Randomized algorithm

We give here a basic way to extract the PSD vectors: Choose randomly m time indexes : $t_1, t_2, \ldots, t_m$ and use the "local mean" of $g_{t_k}(f)$ as a PSD.

Algorithm 1

1: Choose randomly m time indexes : $t_1, t_2, \ldots, t_m$.
2: Set $\sigma_k^2(f) \propto \sum_{\tau=-d}^{+d} g_{t_k+\tau}(f)$, where d is a small integer.

2.2. Correlation based algorithm

The present method uses a correlation measure $\Gamma(g_{t_i}, g_{t_j})$ in order to group similar spectral vectors $g_t(f)$.

Algorithm 2

1: Initialize the classes $C^1, \ldots, C^m$ by filling them randomly with all the data.
2: For all g_{t_i}, compute the score $\alpha_i(k) = \text{mean}_{g \in C^k} \Gamma(g, g_{t_i})$ or $\alpha_i(k) = \text{median}_{g \in C^k} \Gamma(g, g_{t_i})$, as a function of the class k.
3: Form the new classes based on :

$$C^k = \{g_{t_i} | \max_{c=1,\ldots,m} \alpha_i(c) = k\}.$$

4: Goto 2, until convergence.

We use then the following formula

$$\sigma_k^2(f) = \text{mean}\{g_{t_i}(f)\}_{(i \in C^k)} \text{ or } \text{median}\{g_{t_i}(f)\}_{(i \in C^k)}$$

2.3. Additive mixture based algorithm

We use the algorithm for the learning of additive representation exposed in [3] which can be justified in a Bayesian formalism [4].

Algorithm 3

Repeat until convergence
1: Compute the parameters $a_k(t)$ for a given PSD set $\{\sigma_k^2(f)\}_{k \in K}$ and for given data samples $s(t_1), \ldots, s(t_N)$, with any sparse, non negative, decomposition algorithm.
2: Update the PSD set

$$\sigma_{k \text{ new}}^2(f) = \sigma_{k \text{ old}}^2(t) - \mu \sum_l \sigma_{l \text{ old}}^2(t) \sum_t a_l(t) a_k(t) + a_k(t) - a_l(t)$$

The different algorithms above will be compared in section 4. We now study a method for the decomposition of a spectral vector $|\mathcal{S}x(t,f)|^2$ on a PSD set (as needed at step 2 in the general framework).

3. SPARSE, NON NEGATIVE DECOMPOSITION METHOD

In this section, we look for a decomposition algorithm $x \approx \Sigma a, a \geq 0$, where $x = [|Sx(f,t)|^2], a = [a_1(t), \ldots, a_m(t)]$ and $\Sigma = [\sigma_1^2(f), \ldots, \sigma_m^2(f)]$.
We optimize the following criterion

$$\min_{a \geq 0} \frac{1}{2} \|\Sigma a - x\|_2^2 + f(a) \qquad (1)$$

f is a penalty function of the form : $f(a) = \gamma \sum_i a_i^\alpha$ with $\alpha \leq 1$, γ being a sparsity parameter.
In the context of unconstrained optimization, this penalty function leads to sparse solutions [5], that is to say solutions with few non zero coefficients a_i.
For the penalized problem, $\forall i \; a_i \geq 0$, we introduce the Lagrange functional

$$L(a, \lambda) = \frac{1}{2} \|\Sigma a - x\|_2^2 + \gamma \sum_i a_i^\alpha - \sum_i \lambda_i a_i$$

Where $\lambda_i \geq 0$ are the Lagrange multipliers.
The Lagrange functional may be re-written

$$L(a, \lambda) = \frac{1}{2} \|\Sigma a - x\|_2^2 + a^T G(a, \lambda) a$$

where $G(a, \lambda) = \text{diag}\left[\frac{\gamma a_i^\alpha - \lambda_i a_i}{a_i^2}\right]$.

This remark leads to an iterative scheme formulation
Suppose that we are given an estimate $(a^{(l)}, \lambda^{(l)})$ of the optimal solution of (1). Then, we may improve the estimate

by replacing $G(a,\lambda)$ with $G^{(l+1)} = G(a^{(l)}, \lambda^{(l)})$ and minimize the Lagrange functional which is now a quadratic form of a.
We get a new estimate : $a^{(l+1)} = \left[\Sigma^T\Sigma + G^{(l+1)}\right]^{-1}\Sigma^T x$

Remains the evaluation of the Lagrange multipliers $\lambda_i^{(l+1)}$. As we have $\partial L(a,\lambda)/\partial\lambda_i = -a_i$ and λ_i must be positive, we use the following gradient ascent method (Uzawa algorithm, [6]). That is $\lambda_i^{(l+1)} = \max\left(\lambda_i^{(l)} - \nu_{l+1}a_i^{(l+1)}, 0\right)$. ν_l is the learning rate at step l.
Thus we get the following iterative algorithm

Sparse non negative representation algorithm 1

1: Initialize $a^{(0)} = \left(\Sigma^T\Sigma + \beta I\right)^{-1}\Sigma^T x$.
2: repeat until convergence (step $l+1$)

1. $G^{(l+1)} = G(a^{(l)}, \lambda^{(l)})$
2. $a^{(l+1)} = \left[\Sigma^T\Sigma + G^{(l+1)}\right]^{-1}\Sigma^T x$
3. $\lambda_i^{(l+1)} = \max\left(\lambda_i^{(l)} - \nu_{l+1}a_i^{(l+1)}, 0\right)$

Finally, as the matrix inversion in step 2 of each iteration may be prohibitive, we use as a variant of the algorithm with a scaled gradient descent ([6]) in step 2. After simplification, we get

Sparse non negative representation algorithm 2

1: Initialize $a^{(0)} = \left(\Sigma^T\Sigma + \beta I\right)^{-1}\Sigma^T x$.
2: repeat until convergence

1. $g_i^{(l+1)} = \frac{\{a_i^{(l)}\}^2}{\gamma\{a_i^{(l)}\}^\alpha - \lambda_i^{(l)}a_i^{(l)}}$
   $f_i^{(l+1)} = \frac{g_i^{(l+1)}}{\beta g_i^{(l+1)}+1}$, $e_i^{(l+1)} = \frac{a_i^{(l)}}{\beta g_i^{(l+1)}+1}$
2. $a^{(l+1)} = a^{(l)} - \mu_{l+1}\left[f^{(l+1)}\cdot\Sigma^T(\Sigma a^{(l)} - x) + e^{(l+1)}\right]$
3. $\lambda_i^{(l+1)} = \max\left(\lambda_i^{(l)} - \nu_{l+1}a_i^{(l+1)}, 0\right)$

4. EXPERIMENTAL STUDY

4.1. Experimental protocol

We have tested the proposed general framework for a mixture of two audio sources : an audio excerpt from the first "suite" for cello by J.S. Bach (s_1) and an audio excerpt from an African drums piece by Saint Pierre (s_2). The pieces are sampled at 11kHz and we use a window of length 512 samples ($\approx$ 47ms), for the STFT. Note that the sources are decorrelated (i.e $\frac{|\langle s_1, s_2\rangle|}{\|s_1\|\|s_2\|} \approx 0.006$).
We use the one minute of both excerpts as training parts (learning the PSD sets), and the next 15 seconds of both sources are added to form the mixture, in which the sources will be estimated.

4.2. Evaluation criteria

In the experiments, we have the original sources s_1 and s_2 and their estimates $\hat{s}_1$ and $\hat{s}_2$.

Let us use the projection of the estimated sources over the vector space spanned by the real sources.
We may write $\hat{s}_1 = \alpha_1 s_1 + \alpha_2 s_2 + n_1$ and $\hat{s}_2 = \beta_1 s_1 + \beta_2 s_2 + n_2$.
Then we define the source to interference ratio (SIR) and the source to artefact ratio (SAR) (in dB)

$$\mathrm{SIR}_1 = 20\log\left|\frac{\alpha_1}{\alpha_2}\right|\frac{\|s_1\|}{\|s_2\|} \quad \mathrm{SAR}_1 = 20\log\frac{\|\hat{s}_1 - n_1\|}{\|n_1\|}$$
$$\mathrm{SIR}_2 = 20\log\left|\frac{\beta_2}{\beta_1}\right|\frac{\|s_2\|}{\|s_1\|} \quad \mathrm{SAR}_2 = 20\log\frac{\|\hat{s}_2 - n_2\|}{\|n_2\|}$$

The SIR is a way to measure of the residual of the other source in the estimation of each source, whereas SAR score is an estimate of the amount of distortion in each estimated signal.

4.3. Evaluation

We evaluate the scores with varying numbers of PSD vectors (Card(K_i), $i = 1, 2$) for each source, between 5 and 30. In tables 1 and 2, we have the same number of PSD patterns for the two sources. The figures are the SIR and the SAR for both estimated sources.
Note that we have used $\gamma = 10^{-6}$, $\alpha = 1$ for the sparsity parameters in the decomposition method. Indeed, in the experiments, the sparsity is already enforced by the low number of vectors.

The SAR are globally lower than the SIR. This may be intrinsic to the Wiener filtering method, as we do not estimate the exact phases of both source, but take the one of the mixture in both cases. This is may be a limitation of the source model, which is phase independent.
We can also note that both algorithms 2 and 3 (correlation-based and mixture-based) perform better than the plain randomized algorithm.
Note the scores for the randomized algorithm have been averaged over 80 runs.

In tables 1 and 2, the ratios for the drum source get better, as the number of PSD vectors increases, whereas they get worse for the cello source.
Therefore, we have taken, in the other two tables, 5 PSD vectors for the cello and 15, 20 or 30 PSD vectors for the drums.
The best scores are obtained by the second algorithm (correlation-based) with 15 PSD vectors for the drums. Note that the ratios of the standard Wiener filtering are 11.1 (cello), 11.4

(drums) for the SIR, and 6.8 for both SAR.
This suggests that there is an optimal dimensionality of PSD set for each of the sources, in the separation context. This is revealed by the ratio values with varying number of PSD vectors.
Consequently, the sparsity criterion may be further elaborated. The scores would seemingly be increased if we could use a criterion on the exact number of active components in the decomposition method in step 2 of the general framework.

# state	source	random	correlation based	mixture based
5	cello	12.2	13.8	13.7
	drums	12.7	15.8	15.9
10	cello	11.8	12.0	12.3
	drums	15.3	15.6	15.5
30	cello	12.1	11.6	11.0
	drums	18.4	17.8	17.0

TAB. 1 –. SIR for each of the sources as a function of the number of PSD vectors for each source and of the construction method of those vectors

# state	source	random	correlation based	mixture based
5	cello	5.0	6.5	6.6
	drums	5.0	6.2	6.2
10	cello	5.9	6.5	6.5
	drums	5.2	5.8	5.9
30	cello	6.3	7.0	7.7
	drums	5.0	6.0	6.6

TAB. 2 –. SAR for each of the sources as a function of the number of PSD vectors for each source and of the construction method of these vectors

# state	source	random	correlation based	mixture based
5	cello	14.2	15.6	15.8
30	drums	10.5	12.6	12.2
5	cello	14.1	15.5	15.2
20	drums	10.9	12.5	12.6
5	cello	14.2	15.0	15.1
15	drums	11.3	15.0	12.6

TAB. 3 –. SIR for each of the sources as a function of the number of PSD vectors for each source and of the construction method of these vectors

# state	source	random	correlation based	mixture based
5	cello	4.6	6.0	5.8
30	drums	5.4	6.6	6.5
5	cello	4.6	5.8	5.7
20	drums	5.4	6.4	6.4
5	cello	4.9	6.6	5.8
15	drums	5.6	6.6	6.3

TAB. 4 –. SAR for each of the sources as a function of the number of PSD vectors for each source and of the construction method of these vectors

5. CONCLUSION

We have proposed a new method for separation of two sound sources from a single sensor. This is a generalization of the Wiener filtering with locally stationary, non gaussian, parametric source models. We have studied three algorithms for the learning phase and we provide a sparse non negative representation algorithm for the separation phase. On the tests on real data, the method gives very relevant results.

6. REFERENCES

[1] N. Wiener, *Extrapolation, interpolation and smoothing of stationary time series*, MIT press, 1949.

[2] J. Portilla, V. Strela, M.J. WainWright, and E. Simoncelli, "Adaptive wiener denoising using a gaussian scale of mixture model in the wavelet domain," in *Proc. of the 8th international onference on Image Processing*, Thessaloniki,Greece, October 2001.

[3] Michael S. Lewicki and Terrence J. Sejnowski, "Learning overcomplete representations," *Neural Computation*, vol. 12, no. 2, pp. 337–365, 2000.

[4] Olivier Bermond and Jean-François Cardoso, "Approximate likelihood for noisy mixtures," in *Proc. ICA '99, Aussois, France*, 1999, pp. 325–330.

[5] K. Kreutz-Delgado, B.D. Rao, and K. Engan, "Convex/schur-convex (csc) log-priors and sparse coding," in *6th Joint Symposium on Neural Computation*, 1999.

[6] D.P. Bertsekas, *Nonlinear Programming, second edition*, MIT, 1999.