

Double Sparsity: Towards Blind Estimation of Multiple Channels

Prasad Sudhakar, Simon Arberet, Rémi Gribonval

► **To cite this version:**

Prasad Sudhakar, Simon Arberet, Rémi Gribonval. Double Sparsity: Towards Blind Estimation of Multiple Channels. Vincent Vigneron and Vicente Zarzoso and Eric Moreau and Remi Gribonval and Emmanuel Vincent. Latent Variable Analysis and Signal Separation, 9th International Conference on (LVA/ICA2010), Sep 2010, St Malo, France. Springer, 6365 (6365), pp.571–578, 2010, Lecture Notes on Computer Science. <inria-00537756>

HAL Id: inria-00537756

<https://hal.inria.fr/inria-00537756>

Submitted on 22 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Double Sparsity: Towards Blind Estimation of Multiple Channels

Prasad Sudhakar¹, Simon Arberet² and Rémi Gribonval¹

¹ METISS Team, Centre de recherche INRIA Rennes - Bretagne Atlantique
Rennes CEDEX 35042, France
`{firstname.lastname}@inria.fr`

² Institute of Electrical Engineering
École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne, Switzerland
`{firstname.lastname}@epfl.ch`

Abstract. We propose a framework for blind multiple filter estimation from convolutive mixtures, exploiting the time-domain sparsity of the mixing filters and the disjointness of the sources in the time-frequency domain. The proposed framework includes two steps: (a) a clustering step, to determine the frequencies where each source is active alone; (b) a filter estimation step, to recover the filter associated to each source from the corresponding incomplete frequency information. We show how to solve the filter estimation step (b) using convex programming, and we explore numerically the factors that drive its performance. Step (a) remains challenging, and we discuss possible strategies that will be studied in future work.

Key words: blind filter estimation, sparsity, convex optimisation

1 Introduction

Source separation systems have several applications such as speech processing, music transcription, biomedical signal processing, etc. In a general setting, we consider M mixtures $x_i(t), i = 1 \dots M$ of N source signals $s_j(t), j = 1 \dots N$, given by the convolutive model

$$x_i(t) = \sum_{j=1}^N (a_{ij} \star s_j)(t) + v_i(t) \quad (1)$$

where $a_{ij}(t)$ is a filter of length L which models the impulse response between the j^{th} source and the i^{th} microphone, and $v_i(t)$ is the noise at the i^{th} microphone. For brevity, we denote the sources, filters, noise and mixtures by s_j, a_{ij}, v_i and x_i respectively, by dropping the time index.

Blind source separation (BSS) systems attempt to estimate the sources given only the mixtures. This is often done in two stages: the mixing filters are estimated first, and subsequently they are used for source estimation. In case of instantaneous and anechoic mixtures, the filters are simply scalars or time-delayed

scalars and several methods [1] (and references within) have been proposed to estimate the mixing parameters. Many methods such as DUEE [2] rely on the sparsity and disjointness of the sources in a transform domain to estimate the parameters and the sources.

The problem gets more complicated with convolutive mixtures. Frequency domain techniques transform convolutive mixtures problem into multiple complex-valued instantaneous mixtures problem, under the narrowband approximation. But, this approach suffers from the ambiguities of arbitrary scaling and permutations of the sources and solving them is a challenging problem in itself [3].

On the other hand, when there is only one source, the problem of blindly estimating filters from the filtered versions of the source is a well studied problem. In addition, if the filters are sparse then the filter estimation problem can be cast as a standard sparse vector recovery problem [4]. Subsequently, sparse recovery algorithms can be used to solve it.

In a nutshell, there are techniques exploiting source sparsity to estimate instantaneous and anechoic mixing parameters and there are techniques to estimate sparse filters blindly in a single source setting. The grand goal of our effort is to combine these two strands of work and propose a blind mixing filter estimation framework which exploits the time-frequency domain source sparsity and time domain filter sparsity simultaneously. The proposed framework involves a source activity estimation step (clustering) and a filter estimation step.

Ideally, in such a framework the clustering step has to be performed blindly using the mixtures, and the filter recovery process depends on this clustering step. However, as the contribution in this paper, we focus on the formulation and experimental validation of the filter recovery step by solving the clustering step with strong side information. This serves as the first step in realising a completely blind system.

2 Sparse filter estimation

Before we present our contributions, let us first describe some existing work on blind estimation of sparse filters in single and multiple sources settings.

Case of a single source: Let us start with the simplest case of estimating filters when there is only one source s and two outputs x_1 and x_2 . This is the single-input-two-output (SITO) case and we have: $x_i = a_i \star s + v_i, i = 1, 2$. Consider a single frame of s of length T and the length of the filters be L , then the length of x_i will be $T + L - 1$. In the absence of noise, we have the following cross-relation (CR) [5].

$$x_2 \star a_1 = x_1 \star a_2 \tag{2}$$

For convenience, let us associate the signal a_i to the column vector $\mathbf{a}_i = [a_i(t)]_{t=1}^L$ and likewise s to \mathbf{s} and x_i to \mathbf{x}_i .

The convolution $x_i \star a_j$ is associated to the multiplication between the Toeplitz matrix³

$$\mathcal{T}[\mathbf{x}_i] = \begin{bmatrix} x_i(0) & 0 & \cdots & 0 \\ x_i(1) & x_i(0) & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ x_i(T+L-3) & \cdots & \cdots & x_i(0) \end{bmatrix}, \quad (3)$$

and the vector \mathbf{a}_j . By using the shorthand $\mathcal{B}[\mathbf{x}_1, \mathbf{x}_2] = [\mathcal{T}[\mathbf{x}_2], -\mathcal{T}[\mathbf{x}_1]]$, we can write the CR (2) as

$$\mathcal{B}[\mathbf{x}_1, \mathbf{x}_2] \cdot \mathbf{a} = \mathbf{0}, \text{ where } \mathbf{a} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \end{bmatrix}. \quad (4)$$

This relation has inspired several methods (named CR methods) to estimate the filters blindly from the observations [5]. These methods generally do not assume anything about the nature of the filters, however in scenarios such as underwater/wide band wireless communications, the filters that model the channels are often sparse in the time domain.

With the additional sparsity assumption, the SITO filter estimation problem can be formulated as the following ℓ^1 minimisation problem [4], with $\mathbf{B} := \mathcal{B}[\mathbf{x}_1, \mathbf{x}_2]$.

$$\text{minimize } \|\mathbf{a}\|_1 \quad \text{subject to } \|\mathbf{B} \cdot \mathbf{a}\|_2 \leq \epsilon \text{ and } \|\mathbf{a}\|_2 = 1. \quad (5)$$

The normalisation $\|\mathbf{a}\|_2 = 1$ mentioned in [4] is to avoid the trivial zero-vector solution. However, this makes the problem non-convex, and there remains a shift ambiguity of the solution. As an alternative, we use the constraint $a_1(t_0) = 1$, where t_0 is an arbitrarily chosen time index. The resulting problem is convex:

$$\text{minimize } \|\mathbf{a}\|_1 \quad \text{subject to } \|\mathbf{B} \cdot \mathbf{a}\|_2 \leq \epsilon \text{ and } \mathbf{a}_1(t_0) = 1 \quad (6)$$

It can be solved using any standard convex optimisation algorithm, and we chose to use the CVX software package [8].

Case of multiple sources: When dealing with multiple sources, the CR formulation (2) cannot be directly used without further assumptions. Aïssa-El Bey *et al.* [6] have extended the above described SITO approach to N sources, by assuming that it is possible to identify time segments where only one source contributes to the mixtures. Then a SITO problem can be formulated locally at such segments and solved to obtain the filters for that particular source.

Let us describe this with an illustration. Fig. 1(a) is the time-domain plot of two sources s_1 and s_2 . Fig. 1(b) shows the plot of their mixtures, x_1 and x_2 , obtained by convolving the sources with the filters $a_{ij}, i = 1, 2$. These mixtures do not satisfy the CR (4) over the entire time frame, but the sources are such that at the time interval I_j , only source j is active. If we extract the mixtures at

³ Calligraphic letters will denote operators that map a vector to a matrix, e.g. $\mathcal{T}[\mathbf{x}_i]$.

the appropriate time segments \tilde{I}_j , then we obtain the segments of the mixtures $y_i^{(j)} = \{x_i(t)\}_{t \in \tilde{I}_j}$ that depend on a single source j : $y_i^{(j)} = a_{ij} \star \tilde{s}_j$, where \tilde{s}_j is the restriction of the source to a certain time interval. The vectors $\mathbf{y}_i^{(j)}$ corresponding

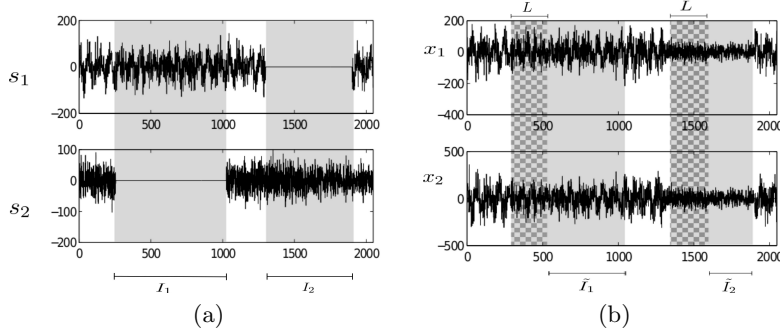


Fig. 1. (a) Sources with intervals where only one source is active. (b) Mixtures from the sources.

to $y_i^{(j)}$ now satisfy the CR $\mathcal{B}[\mathbf{y}_1^{(j)}, \mathbf{y}_2^{(j)}] \cdot \mathbf{a}^{(j)} = \mathbf{0}$ and so this can be used to estimate the filters for source j by solving the optimisation problem (6) with $\mathbf{B} = \mathcal{B}[\mathbf{y}_1^{(j)}, \mathbf{y}_2^{(j)}]$. The authors of [6] have explicitly presented a technique to identify the intervals \tilde{I}_j and have experimentally demonstrated the results of this approach.

3 Proposed framework

In general, the sources may overlap in the time domain, and the approach described in the previous section might not be suitable for the filter estimation task, even if the filters are sparse. Instead of disjoint time supports, it is a common assumption in BSS to consider sources with almost disjoint time-frequency (T-F) representations, in the short-time Fourier transform (STFT) domain.

T-F domain SITO: Let us start with the single source setting again. Let $\hat{\mathbf{x}}_i$ be the STFT of the vector \mathbf{x}_i , and $\hat{\mathbf{a}}_i$ be the Fourier transform of \mathbf{a}_i (appropriately zero padded) such that $\hat{\mathbf{x}}_i = [\hat{\mathbf{x}}_i(\tau, f)]_{\tau, f}$ and $\hat{\mathbf{a}}_i = [\hat{\mathbf{a}}_i(f)]_f$. Let us consider STFT frames $1 \leq \tau \leq N_T$. In each frame, the cross relation (2) becomes

$$\hat{\mathbf{x}}_2(\tau, f) \cdot \hat{\mathbf{a}}_1(f) \simeq \hat{\mathbf{x}}_1(\tau, f) \cdot \hat{\mathbf{a}}_2(f), \quad \forall f. \quad (7)$$

Defining $\mathcal{D}[\hat{\mathbf{x}}_i](\tau) = \text{diag}([\hat{\mathbf{x}}_i(\tau, f)]_f)$, the CR in the STFT domain will be

$$\hat{\mathcal{B}}[\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2] \cdot \mathbf{a} \simeq \mathbf{0} \text{ with } \hat{\mathcal{B}}[\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2] = \begin{bmatrix} \mathcal{D}[\hat{\mathbf{x}}_2](1), & -\mathcal{D}[\hat{\mathbf{x}}_1](1) \\ \vdots & \vdots \\ \mathcal{D}[\hat{\mathbf{x}}_2](N_T), & -\mathcal{D}[\hat{\mathbf{x}}_1](N_T) \end{bmatrix} \begin{bmatrix} \mathbf{F}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{F}^* \end{bmatrix}, \quad (8)$$

where \mathbf{F}^* is the Fourier matrix of appropriate size. Using this CR, the optimisation problem (6) with $\mathbf{B} := \widehat{\mathcal{B}}[\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2]$ can be solved to obtain the filters.

Multiple sources: In the case of N sources, the above formulation can be generalised. In the time domain, the intervals I_j enabled us to formulate the CR for each source. Likewise, if we can identify a set of T-F points Ω_j for each source j , where only one source is active, then these sets can be used to formulate the CR and estimate the filters as described previously. For each source j , one can build a matrix $\widehat{\mathcal{B}}_{\Omega_j}[\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2]$ that contains the rows of $\widehat{\mathcal{B}}[\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2]$ indexed by the T-F points $(\tau, f) \in \Omega_j$ and form the cross relation $\widehat{\mathcal{B}}_{\Omega_j}[\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2] \cdot \mathbf{a}^{(j)} = 0$. Then for each source j , we can estimate the filter $\mathbf{a}^{(j)}$ by solving (6) with $\mathbf{B} := \widehat{\mathcal{B}}_{\Omega_j}[\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2]$.

Proposed framework: The mixing filter estimation process using time-frequency domain cross relation can be summarized in the following steps.

-
1. Compute the time-frequency representations $\widehat{\mathbf{x}}_i$, $i = 1, 2$.
 2. For each source j ,
 - P1** { (a) Identify the set Ω_j .
 - P2** { (b) Build the matrix $\mathbf{B} := \widehat{\mathcal{B}}_{\Omega_j}[\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2]$.
 - (c) Solve (6) to obtain the estimated filter $\tilde{\mathbf{a}}^{(j)}$.
-

In the rest of this paper, we focus on the evaluation of the performance of the proposed framework *when the solution to P1 is known*. Further work will be devoted to addressing **P1** and **P2** simultaneously, and preliminary ideas will be discussed in the conclusion.

4 Experimental verification

The framework presented in the previous section was experimentally verified in the multiple source setting for $N = 2$ and $N = 3$. The paragraphs below present the details of data generation, the experimental protocol and the results.

4.1 Data generation

Filters: The length of the individual filters \mathbf{a}_i was set to $L = 256$, and their sparsity was set to $\|\mathbf{a}_i\|_0 = k/2$, for various values of k . So the unknown vector \mathbf{a} was of length $2 * L = 512$ and its sparsity was k . The $k/2$ support indices on each channel were chosen uniformly at random in the set $(\frac{L}{4}, \frac{3L}{4})$. The filter coefficients were generated i.i.d. Gaussian with zero mean, unit variance and sorted to have decreasing magnitudes along the time axis within the support. Every filter $a_{1j}(t)$ was finally normalised and shifted to have $a_{1j}(L/2) = 1$.

Sources: For each source j , we generated N_T independent time frames $s_j^\tau(t)$, $1 \leq \tau \leq N_T$. Each frame $s_j^\tau(t)$ of length $T = 3 * L$ was a sum of N_F sinusoids:

$$s_j^\tau(t) = \sum_{w=1}^{N_F} A_{jw}^\tau \sin(2\pi f_{jw}^\tau t + \phi_{jw}^\tau), \quad (9)$$

where the frequencies f_{jw}^τ were chosen uniformly at random in $[0, 1/2]$. The amplitudes A_{jw}^τ were generated i.i.d. Gaussian with zero mean, unit variance, and the phases ϕ_{jw}^τ were chosen uniformly at random in $[0, 2\pi]$.

Performance: By nature, the estimated solution $\tilde{a}^{(j)}(t)$ suffers from shift and scaling ambiguity (to satisfy the normalisation constraint). The following definition of the output signal-to-noise ratio (SNR), which accounts the scaling and shift ambiguity, was used as a recovery performance measure.

$$\text{SNR}_{out} = 10 \log_{10} \left(\frac{\sum_j \sum_t \|a^{(j)}(t)\|_2^2}{\min_{t', \mu} \sum_j \sum_t (\|a^{(j)}(t) - \mu \cdot \tilde{a}^{(j)}(t - t')\|_2^2)} \right). \quad (10)$$

4.2 Experimental protocol

We performed experiments in the ideal single source setting for various values of filter sparsity k and number of sinusoids per frame N_F .

We determined the number of frames required to recover the filters with an output SNR (defined above) of 20dB. This number depends on the sparsity k and the number of sinusoids per frame N_F , and let us denote this by $\#_{20}(k, N_F)$. In the experiments for multiple source setting, we arbitrarily chose $N_T(k, N_F) = \#_{20}(k, N_F) \times 2$.

For every combination of k and N_F , 20 independent trials were done and the performance was averaged. In each trial, the sources and filters were generated as described previously, the mixtures x_i were obtained according to (1) with $v_i = 0$ and the vectors $\hat{\mathbf{x}}_i$ were formed. For each source j , **P1** and **P2** were solved as described below.

P1: Obtaining Ω_j using side information: The set Ω_j was constructed using as side information the frequencies $\{f_{jw}^\tau\}_{w=1}^{N_F}$ that are used to generate the source in (9). Let $\hat{\mathbf{s}}_j^\tau$ be the time-frequency domain vector of length $F = T + L - 1$ obtained by the appropriate zero-padding and transformation of the frame τ of the source s_j^τ . We defined, using $\theta = 10\text{dB}$,

$$(\tau, f) \in \Omega_j \iff f \in \{f_{jw}^\tau\}_{w=1}^{N_F} \text{ and } 20 \cdot \log_{10} \left(\frac{|\hat{\mathbf{s}}_j^\tau(f)|}{|\hat{\mathbf{s}}_{j'}^\tau(f)|} \right) \geq \theta, \forall j' \neq j. \quad (11)$$

P2: Filter estimation by convex optimization: For each source j , the matrix $\tilde{\mathcal{B}}_{\Omega_j}[\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2]$ was built using the set Ω_j . Then, the resulting convex optimisation problem (6) was solved to obtain the filter estimates $\tilde{\mathbf{a}}^{(j)}$. The value of

ϵ used in (6) actually tells us about the amount of imperfection that we would like to allow in the CR. Setting ϵ is a challenging task. In these experiments, we relied upon an oracle setting of ϵ . For each source j , we used $\epsilon_j = \|\mathbf{B} \cdot \mathbf{a}^{(j)}\|_2$, with $\mathbf{a}^{(j)}$ the true filter.

Results: Tables 1 and 2 show the average output SNR for various (k, N_F) for 2 and 3 sources respectively. In both cases, the anechoic filters are recovered with very high SNR, and the output SNR is at least 10dB when $N_F \leq 3$ and $k \leq 10$. For a given sparsity k , the output SNR drops as the number of sinusoids per frames N_F increases. We experimented with higher values of N_F and we found that the performance continues to degrade. This is because the sources tend to interfere more as N_F increases, thereby violating the CR badly. Indeed, even though we generated sums of sinusoids, their Fourier transform has peaks at the associated frequencies that can have a large main lobe and secondary lobes, leading to interferences. This could be compensated by setting a higher threshold θ to compute the set Ω_j , at the price of a smaller number of “visible” frequencies per time frame, which in turn could be compensated by increasing the number of observed time frames $N_T(k, N_F)$.

5 Discussion and future work

We have described the existing work on the time-domain CR method to estimate sparse filters from convolutive mixtures. The method exploits the time domain disjointness of the sources, which is rather a restrictive scenario. By making a more realistic assumption that the sources are disjoint in the time-frequency domain, we have extended the CR formulation to the time-frequency domain and proposed a framework to estimate sparse filters. The framework contains a time-frequency clustering step followed by a filter estimation step. In a setting where the clustering is performed using strong side information, we have presented the results of experimental evaluation of the filter estimation step.

In the future, we would like to understand how to set ϵ with less or no prior information. As mentioned earlier, the recovery performance could be improved by using a higher threshold θ and a larger number of time frames $N_T(k, N_F)$. Also, the run-time of the algorithm for large problem sizes is an issue and we would like to explore alternative fast algorithms.

Table 1. Average output SNR for $N = 2$.

	$k = 2$ (Anechoic)	$k = 4$	$k = 6$	$k = 8$	$k = 10$	$k = 12$	$k = 14$	$k = 16$
$N_F = 1$	60.76	29.07	19.24	18.96	18.68	10.95	13.39	13.84
$N_F = 2$	47.98	23.85	18.15	15.51	15.00	9.08	9.51	10.37
$N_F = 3$	46.38	24.99	15.77	17.35	13.98	9.59	12.15	9.79
$N_F = 4$	44.57	20.18	14.84	16.88	14.74	9.18	8.22	9.01
$N_F = 5$	42.30	21.75	11.87	15.37	13.82	10.30	8.21	7.52

Table 2. Average output SNR for $N = 3$.

	$k = 2$ (Anechoic)	$k = 4$	$k = 6$	$k = 8$	$k = 10$	$k = 12$	$k = 14$	$k = 16$
$N_F = 1$	52.54	26.34	20.74	20.64	17.79	13.41	11.76	13.05
$N_F = 2$	51.49	23.85	16.89	16.99	15.30	9.95	10.49	9.28
$N_F = 3$	44.91	22.74	13.36	13.62	15.04	10.87	10.51	8.60
$N_F = 4$	44.00	21.34	13.73	13.61	10.98	10.05	9.38	8.91
$N_F = 5$	39.68	23.50	13.52	13.41	11.21	8.97	9.04	7.60

We will also consider solving the clustering and filter estimation problems simultaneously. Given an estimate of the filters, we would like to formulate and solve a convex problem to accomplish the clustering task. We intend to estimate the clusters and filters by solving the corresponding convex problems alternatively.

Acknowledgements

This work was supported in part by Agence Nationale de la Recherche (ANR), project ECHANGE (ANR-08-EMER-006) and by the EU FET-Open project FP7-ICT-225913-SMALL.

References

1. S. Arberet, R. Gribonval and F. Bimbot. A Robust Method to Count and Locate Audio Sources in a Multichannel Underdetermined Mixture. *IEEE Trans. on Signal Processing*, (2010) vol. 58 (1) pp. 121 - 133.
2. A. Jourjine, S. Rickard and O. Yilmaz. Blind separation of disjoint orthogonal signals: Demixing n sources from 2 mixtures. *ICASSP 2000*, vol. 5, pp. 2985-2988
3. M. S. Pedersen, J. Larsen, U.Kjems and L. C. Parra. A survey of convolutive blind source separation methods. *Springer Handbook of Speech Processing*, Springer, New York, 2007.
4. A. Aïssa-El-Bey and K. Abed-Meraim. Blind SIMO channel identification using a sparsity criterion. *SPAWC 2008*, pp. 271 - 275.
5. H. Liu, G. Xu and L. Tong. A deterministic approach to blind identification of multi-channel FIR systems. *ICASSP-94.*, vol. 4, pp. 581 - 584.
6. A. Aïssa-El-Bey, K. Abed-Meraim, and Y. Grenier. Blind Separation of Underdetermined Convolutive Mixtures Using Their Time-Frequency Representation. *IEEE TASLP*, Vol. 15, No. 5, July 2007, pp. 1540-1550.
7. E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Info. Theory*, Vol. 52, No. 2, Feb. 2006, pp. 489-509.
8. M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming (web page and software). <http://stanford.edu/~boyd/cvx>, June 2009.