Interference reduction on full-length live recordings
Diego Di Carlo, Antoine Liutkus, Ken Déguernel

To cite this version:


HAL Id: hal-01713889
https://inria.hal.science/hal-01713889
Submitted on 21 Feb 2018

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.
ABSTRACT

Live concert recordings consist in long multitrack audio samples with significant interferences between channels. For audio engineering purposes, it is desirable to attenuate those interferences. Recently, we proposed an algorithm to this end based on Non-negative Matrix Factorization, that iteratively estimate the clean power spectral densities of the sources and the strength of each in each microphone signal, encoded in an interference matrix. Although it behaves well, this method is too demanding computationally for full-length concerts lasting more than one hour. In this paper, we show how random projections of the data can be leveraged for effective estimation of the parameters. Interference reduction with these ideas can be achieved on full-length live multi-track recordings in an acceptable time and could be used by sound engineers. We demonstrate the efficiency of this approach on real full-length live recordings from the Montreux Jazz Festival and also provide an implementation of the method.

Index Terms— interference reduction, microphone leakage, source separation, random projection, compressive sensing

1. INTRODUCTION

It is common for musicians and sound engineers to record different instruments at different times. The major benefit of this practice is to obtain clean and isolated tracks that can easily be processed individually in a second stage. However, this practice hinders musical spontaneity and interaction. In some scenarios, such as live performances, orchestral recordings and so on, all the musicians play together. Even if each one is recorded by its own dedicated microphone, spurious sounds are captured as well, such as the sound of other instruments. These are called interferences, or microphone leakage in the sound engineering parlance. As long as the musicians do play in the same room, they cannot be avoided, but only reduced to some extent through sophisticated microphoning practices.

In the last 10 years, many studies have been conducted on the topic of interference reduction both in time domain [3,4] and in time-frequency domain [5,6,2]. Notably, the method described in [2] totally neglects phase dependencies to adopt an energy-based model. More specifically, inspired by the Non-negative Matrix Factorization (NMF [7]), its parameters are twofold. First, the Power Spectral Densities (PSDs) of the sources encode the power spectrum of each source varying along time. Second, the interference matrix specify how much of each source does get into each microphone signal (see Figure 1). After estimation of all parameters, the desired signals are recovered through generalized Wiener filter [8,9].

As such, that approach generalizes [5,6] with the adjunction of the interference matrix and achieves good interference reduction in practice despite its simplifying assumptions.

While being very effective in practice, the method in [2,10] does suffer from two main weaknesses. First, estimation of the sources PSDs relies on ad-hoc heuristics. Second, it cannot scale to full-length recordings due to computational burden. In our previous work [11], we addressed this first issue, showing how a rigorous probabilistic Gaussian framework [8,9,12] may be used to yield provably optimal algorithms. However, the computational problem remains an issue for the actual embedding of such methods in sound engineering devices today. More specifically, the computational bottleneck lies in the estimation of the interference matrix. If it was available, processing could indeed be achieved on a
frame-by-frame basis, strongly reducing computational burden and even allow for real-time processing.

In this work, we show how state-of-the-art performances in interference reduction can be obtained by estimating the interference matrix after application of a dimension reduction method. Such a method should capture as much of the variation of the data as possible \[13\]. While a sensible way to do it would be to use Principal Components Analysis (PCA), random projections were found to be just as effective, as already noticed in \[14\]. The core contribution is to show that the Gaussian framework underlying estimation of the interference matrix still holds in the compressed domain, leading to a drastic decrease of the computational cost of the estimation for similar performance, creating therefore, to our knowledge, the first interference reduction method applicable to the scale of full-length recordings.

2. MODEL AND STATE OF THE ART

2.1. Notation and probabilistic model
We consider \( J \) voices captured by \( I \) microphones, yielding mixtures \( x_i \), with \( i = 1, \ldots, I \). Without loss of generality, every voice \( j = 1, \ldots, J \) is present in all mixtures \( x_i \) and we define the image \( y_{ij} \) as the contribution of voice \( j \) in mixture \( i \). Moving to a Time-Frequency (TF) representation such as the Short Term Fourier Transform (STFT), this translates as:

\[
X_i(f,t) = \sum_{j=1}^{J} Y_{ij}(f,t), \tag{1}
\]

where \( X_i \) and \( Y_{ij} \) are the STFT of \( x_i \) and \( y_{ij} \) and are complex matrices of dimension \( F \times T \), with \( F \) the number of frequency bands and \( T \) the number of time frames. The interference reduction problem consists in computing estimates \( \hat{Y}_{ij} \) for the images \( Y_{ij} \).

We now briefly summarize the assumptions of the Gaussian Framework \[11\][9]. First, all \( \{Y_{ij}(f,t)\}_{ijf} \) are assumed independent. Then, we choose the Local Gaussian Model (LGM, \[15\][9]), adequate for locally stationary signals:

\[
Y_{ij}(f,t) \sim \mathcal{N}_c(0, P_{ij}(f,t)), \tag{2}
\]

where \( \mathcal{N}_c \) is the isotropic complex Gaussian distribution \[16\] and \( P_{ij}(f,t) \geq 0 \) is the Power Spectral Density (PSD) of voice image \( y_{ij} \).

At this point, we model those \( P_{ij} \) by assuming the amount of interference of each voice \( j \) in microphone \( i \) is controlled by a channel-dependent scalar facors \( \lambda_{ij}(f) \geq 0 \) \[2\]:

\[
P_{ij}(f,t) = \lambda_{ij}(f) P_j(f,t), \tag{3}
\]

where \( P_j(f,t) \geq 0 \) is the latent PSD of voice \( j \) and is independent of the channel \( i \) and the \( I \times J \) interference matrix is defined as \( [\Lambda(f)]_{ij} = \lambda_{ij}(f) \) \[2\][11].

As a sum of independent Gaussian variables, \( X_i(f,t) \) is distributed as:

\[
X_i(f,t) \sim \mathcal{N}_c \left( 0, \sum_{j=1}^{J} \lambda_{ij}(f) P_j(f,t) \right), \tag{4}
\]

and the parameters for the model to be estimated are:

\[
\Theta = \left\{ \Lambda(f), \{P_j(f,t)\}_{ij} \right\}. \tag{5}
\]

If these are given, we can estimate any desired \( Y_{ij} \) through generalized Wiener filtering:

\[
\hat{Y}_{ij}(f,t) = \frac{P_{ij}(f,t)}{\sum_{j'=1}^{J} P_{ij'}(f,t)} X_i(f,t) \triangleq W_{i,j}(f,t) X_i(f,t), \tag{6}
\]

where \( W_{i,j}(f,t) \) is called the Wiener gain. Time-domain signals are finally recovered through inverse STFT.

2.2. State of the art
We now briefly review the Music Interference Removal Algorithm (MIRA) for parameter estimation \[11\]. In short, it maximizes the parameters likelihood given the observations \( X_i(f,t) \). This can be shown to be equivalent to:

\[
\hat{\Theta} \leftarrow \arg \min_\Theta \sum_{f,t,i} d_0 \left( V_i(f,t) \parallel \sum_j \lambda_{ij}(f) P_j(f,t) \right), \tag{7}
\]

where \( d_0 \) is the Itakura-Saito divergence \[17\]. In this setting, \( \Lambda(f) \) and \( P_j \) are alternatively updated following the classic Non-negative Matrix Factorization (NMF) methodology:

\[
P_j(f,t) \leftarrow P_j(f,t) \frac{\sum_{i=1}^{I} P_i(f,t) - 2 V_i(f,t) \lambda_{ij}(f)}{\sum_{i=1}^{I} P_i(f,t) - 1 \lambda_{ij}(f)} \tag{8}
\]

\[
\lambda_{ij}(f) \leftarrow \lambda_{ij}(f) \frac{\sum_{i=1}^{I} P_i(f,t) - 2 V_i(f,t) P_j(f,t)}{\sum_{i=1}^{I} P_i(f,t) - 1 P_j(f,t)}, \tag{9}
\]

where \( V_i(f,t) \triangleq |X_i(f,t)|^2 \) and \( P_i(f,t) \triangleq \sum_j \lambda_{ij}(f) P_j(f,t) \). A good initialization to MIRA is to take each \( P_j \) as the spectrogram of the signal captured by the close-microphone for voice \( j \).

3. RANDOM PROJECTION
The MIRA approach presented above is only able to process small-scale data because of its computational load. More precisely, it can be seen that its time and space complexity, using \[8\] and \[9\] is \( O(FTIJ) \). Typical values for a 3 minute long song are \( F = 4096, T = 10000, I = 30, J = 25 \) and computations take about one hour on a typical powerful workstation with 64 Gb of RAM. Processing of a full-length recording is
not tractable in that case and a speed up is required for MIRA to be used by actual sound engineers.

After investigation, the bottleneck of MIRA appears as the updating rule (9) for the interference matrix \( \Lambda \). That update requires a summation over all the \( T \) frames of the recording, forcing the algorithm to access the whole data at each iteration, which cannot be achieved easily in practice. If \( \Lambda(f) \) was known, the algorithm would be significantly faster and may run online. The core idea of the current study is thus to develop a two-stage procedure, where we first estimate the interference matrix \( \Lambda \) in a computationally effective manner and then use it in a second stage to estimate the voices PSD.

The main contribution here then becomes estimation of \( \Lambda \) from a compressed representation of the data, in a very simple instance of compressed learning [13, 18]. The proposed method is to construct a random projection \( M_i(f,r) \) of each recording \( X_i \) of dimension \( F \times R \), with \( R \ll T \). This is done as:

\[
M_i(f,r) \triangleq \sum_{t=1}^{T} X_i(f,t) Q_i(r,t),
\]

where the \( R \times T \) projection matrix \( Q_i \) is composed of independent and identically distributed entries taken as \( Q_i(r,t) \sim \mathcal{N}(0,1) \). Since it is not used in the sequel, this matrix does not need to be stored anywhere in memory but is drawn on-the-go while parsing the data only once to compute \( M_i \).

Thanks to our Gaussian model (4) on the mixtures, we can compute the distribution of \( M_j(f,r) \) as:

\[
M_j(f,r) \sim \mathcal{N} \left( 0, \sum_{ij} \lambda_{ij}(f) S_j(f,r) \right),
\]

with

\[
S_j(f,r) \triangleq \sum_{t} P_j(f,t) Q_j(r,t)^2.
\]

As can be seen, we obtain the same model for \( M_i \) that we had for \( X_i \) in (4), simply replacing \( T \) for \( R \) and \( P_j \) for \( S_j \).

The important point there is that \( \Lambda \) is the same in both cases. Hence, we may learn it along \( S \) from the projections only using MIRA on the projections. This requires summations over \( r \) instead of \( t \) in (9), leading to huge computational savings. Once it is learned, we can discard \( S \) and keep \( \Lambda \) fixed to learn \( P_j \) through (8) on the original data, yielding our proposed fastMIRA.

4. EXPERIMENTAL EVALUATION

In this section, we compare performance and computing time of the proposed fastMIRA method with respect to its original version, MIRA. The two algorithms were run on a whole pop rock multitrack live recording session of Huey Lewis and the News’ song Power of Love at the Montreux Jazz Festival 2000 (length: 5m 10s). This recording features 40 microphones, recording 30 voices. It has a sample-rate of 48 kHz and a depth of 16 bits/sample. The overall size of this multitrack recording is almost 1.2 Gb and was provided by the Montreux Jazz Digital Project and EPFL. To promote reproducibility of the results presented here, we provide an open-source implementation of both MIRA and fastMIRA in the webpage dedicated to this paper.

Since we already performed a thorough perceptual evaluation of MIRA in [11], we decided here to compare the model parameters estimated by both algorithms as a function of the projection dimension \( R = 2^k \), with \( k = 0, 1, \ldots, 13 \). For each \( R \), all the parameters are computed anew. Figure 3a shows the (logarithm of the) reconstruction error (7), which is the cost function minimized by the original MIRA as function of the number of iterations for learning \( P_j \). Thus, that plot quantifies the overall modeling capabilities of fastMIRA as compared to MIRA. In this figure, it is clearly seen that fastMIRA provides similar results than MIRA in terms of reconstruction error after only a few iterations.

However, a similar reconstruction error is not sufficient to conclude about the two algorithms yielding similar model estimates. In fact, it does not take into account \( P_j \) nor \( \Lambda(f) \) separately, but only their product as in (7). To investigate whether fastMIRA does yield the same estimate for \( \Lambda \) than MIRA, we compute the normalized mean square error (nMSE) in logarithm scale between those estimates as a function of \( R \) and the recording length, i.e. the number of frames \( T \). Figure 3b shows the results for this experiment, and we see a clear phase transition between systematic error and a good estimation (nMSE = −1.25 dB), once some threshold value is crossed, say \( R = 64 \). Interestingly, this kind of behaviour

\[\text{See } \text{github.com/Chutlhu/mirapie} \]
is typical for compressed sensing applications. Whatever the number of frames $T$ found in the original recording, we can see that picking $R = 256$ is sufficient to get the same estimate for $\Lambda$ through fastMIRA as the one obtained through MIRA. Once granted that both approaches lead to similar estimates for the model parameters, we informally checked perceptually that they lead to similar separation quality. Doing so, we noticed that fastMIRA seems to provide slightly more isolation in general than MIRA and a bit more distortion. However, this appears to be a very slight effect and we should use a thorough evaluation to really tell whether the two systems can be discriminated perceptually, which is not obvious.

Now, the greatest point of interest in using fastMIRA over MIRA is obviously its reduced computational complexity. Figure 3c shows the time taken for computations on a 8-core computer with 16 Gb of RAM, which is a common setup for a professional sound engineer. We can see that the proposed approach can yield a good approximation in only few minutes, while MIRA takes more than 30 minutes.

Finally, we investigate if the performance of fastMIRA depends on the length of the recordings. To this end, we run the proposed algorithm on two full-length live performances recorded at Montreux Jazz Festival: Huey Lewis and the News (length: 50min; 25 voices) and Sigur Rós (length: 120min, 30 voices). As metrics, we consider the nMSE (log10 scale) between an excerpt (first 5 minutes) from the whole processed concert and the same excerpt processed individually through MIRA. Results are reported in Table 1. From this table, we can notice that the method is not particularly affected by the length of the input recordings. A closer investigation reveals that the provided small amount of distortion depends on the some variability in the stage setup, such as position and usage of the microphones, which happens more often in Sigur Rós performance. Moreover even if the processing time is quite high (17 hours for a 2 hour concert), we want here to highlight that previous method could not even run on recording-studio-like workstation.

<table>
<thead>
<tr>
<th>Recordings</th>
<th>Size [GB]</th>
<th>Duration [min]</th>
<th>Time elapsed [h]</th>
<th>nMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huey Lewis...</td>
<td>10</td>
<td>55</td>
<td>6 h 58 min</td>
<td>0.767</td>
</tr>
<tr>
<td>Sigur Rós</td>
<td>39.2</td>
<td>123</td>
<td>16 h 41 min</td>
<td>0.909</td>
</tr>
</tbody>
</table>

Table 1: Difference in dB between 5-minute of estimated voice images using fastMIRA on the full-length recordings and the same portion processed individually through MIRA. $R = 512$ was used.

5. CONCLUSION

In this paper, we have proposed a simple, yet effective way to reduce the computational load of an algorithm for interference reduction in live multitrack recordings. It is based on exploiting random projections of the input data to estimate model parameters. As we demonstrated, the proposed algorithm is able to achieve estimation and interference reduction just as well as the original method, while going through the data only twice, which is desirable in such massive recordings. During our evaluation, we applied interference reduction on real and challenging multitrack live data from the Montreux Jazz Festival, leading to the first method we are aware of that can reduce voice leakage at this scale. Future work consists in developing a user-friendly interface which can be used by sound engineers to process full-length live recordings. Moreover the impact of having long interference-reduced tracks for Music Information Retrieval tasks will be also studied.

Acknowledgements

This work was made with the support of the French National Research Agency, as part of the projects DYC12 (ANR-14-CE24-0002-01) and KAMoulox (ANR-15-CE38-0003-01). Access to the Montreux Jazz Festival database was provided by EPFL, in the scope of DYC12.
6. REFERENCES


