INFORMED SOURCE SEPARATION VIA COMPRESSIVE GRAPH SIGNAL SAMPLING

Gilles Puy, Alexey Ozerov, Ngoc Q. K. Duong and Patrick Pérez

Technicolor
975 Avenue des Champs Blancs, 35576 Cesson-Sévigné, France

ABSTRACT

We propose a novel informed source separation method for audio object coding based on a recent sampling theory for smooth signals on graphs. Assuming that only one source is active at each time-frequency point, we compute an ideal map indicating which source is active at each time-frequency point at the encoder. This map is then sampled with a compressive graph signal sampling strategy that guarantees accurate and stable recovery at the decoder. The graph is built using feature vectors, computed using non-negative matrix factorization, that allows us to connect similar source activations in the time-frequency plane. We show that the proposed approach performs better than state-of-the-art methods at low bitrate.

Index Terms— Informed source separation, audio object coding, non-negative matrix factorisation, graph signal processing, compressive sampling.

1. INTRODUCTION

As audio source separation has remained a challenging task, especially in the single-channel case (an extreme case of underdetermined situation where the are less observations than unknowns) [1], the use of prior information about the sources to guide the separation process has been largely considered in the literature and is known as informed source separation (ISS) [2]. Such information can be, *e.g.*, music score [3], text transcript [4], or extracted from the sources themselves [5–7]. The latter case concerns audio coding applications in which the so-called *side information* is extracted at the encoding stage where the original sources are known, and then used to guide the source estimation at the decoding stage where only the mixture is observed. It is also related to spatial audio object coding (SAOC), a recent approach standardized in the MPEG audio group, [8] for the same type of practical application. As parametric coding schemes, the encoding processes of both ISS and SAOC require remarkable computation costs at the encoder. Thus, in the same line of research, Bilen *et al.* [9] proposed a compressive sampling-based ISS that shifts the computational load from the encoder to the decoder, making the former extremely fast.

In this paper, we present a novel ISS approach targeting the ability to greatly reduce bitrate for transmission. Motivated by the fact that audio sources are usually disjoint in the time-frequency (TF) representation, *i.e.,* only one source is active at each time-frequency point [10], a map showing which source is active at each time-frequency point is a good indicator to separate the sources from the mixture. At the encoder, as the original sources are known, this oracle map can be easily computed and considered as side information to guide the source separation given the mixture. To compress such side information, this map is then sampled with a compressive graph signal sampling strategy to guarantee the ability to recover it at the decoding stage for source separation. As the underlying graph should be available at both the encoding and decoding sides, we propose to build it using feature vectors derived from the non-negative matrix factorization (NMF) of the mixture signal [11]. Compared to the existing works [5–7, 9], we show that the proposed approach can go toward lower bitrates while still offering reasonable source separation performance at the decoding stage.

The rest of the paper is organized as follows. Section 2 presents the problem formulation, followed by the description of the compression strategy via graph sampling in Section 3. Experimental results are shown in Section 4. Finally, we conclude in Section 5.

2. PROBLEM FORMULATION

In ISS, a mixture of different sources is transmitted to the decoder along with side information to help the separation of the sources from the mixture. We denote by $\boldsymbol{s}_j$, $j = 1, \dots, J$, the different sources in the temporal domain. The mixture $\boldsymbol{x}$ satisfies

$$\boldsymbol{x} = \sum_{j=1}^{J} \boldsymbol{s}_j. \tag{1}$$

At the encoder, our goal is now to construct and transmit additional information that will help the decoder estimate each source $\boldsymbol{s}_j$ from $\boldsymbol{x}$.

A common strategy in audio source separation is to work in the TF domain. The short-time Fourier transform (STFT) of $\boldsymbol{x}$ is computed and it is assumed that a single source is active at each TF point. Under this assumption, extracting source j thus consists in identifying where this source is active in the TF domain. This is the approach we follow to construct the information to transmit for the separation.

Let $\mathsf{X} \in \mathbb{C}^{F \times N}$ and $\mathsf{S}_j \in \mathbb{C}^{F \times N}$ denote the complex matrices of the STFT coefficients of the mixture $\boldsymbol{x}$ and of the source $\boldsymbol{s}_j$, respectively. To determine where source j is active in the TF domain, we compute

$$\mathsf{M}_j = \underset{\mathsf{M} \in \mathbb{R}^{F \times N}}{\operatorname{argmin}} \, \|\mathsf{X} \odot \mathsf{M} - \mathsf{S}_j\|_F^2 = \operatorname{Real}\left(\mathsf{S}_j \oslash \mathsf{X}\right), \tag{2}$$

for each source $j = 1, \dots, J$. In the above equation, $\odot$ and $\oslash$ stand for the element-wise product and division, respectively.[1] We then determine which unique source is dominantly active at each TF point (f, n) by checking which M_j has the largest entry at (f, n). We thus obtain a map $\mathsf{Z} \in \{1, \dots, J\}^{F \times N}$ whose entries are

$$\mathsf{Z}[f, n] \in \underset{1 \leqslant j \leqslant J}{\operatorname{argmax}} \; \mathsf{M}_j[f, n]. \tag{3}$$

Note that, with the above definition, a source index is arbitrarily chosen in the case where all sources are inactive at a TF point.

[1] For simplicity, we assumed that X does not contain any zero entries.

With the knowledge of X and Z, we can estimate all the sources as follows. For each source j, we construct the binary activation matrix $\mathsf{M}_j^* \in \{0,1\}^{F\times N}$ that satisfies

$$\mathsf{M}_j^*[f,n] = \begin{cases} 1 & \text{if } \mathsf{Z}[f,n] = j, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

and compute the inverse STFT of $\mathsf{X} \odot \mathsf{M}_j^*$ to obtain an estimation of $\boldsymbol{s}_j$. Our goal now is to send a compressed version of the map Z to the decoder to be able to estimate the binary masks M_j^* at the decoder.

3. COMPRESSION VIA GRAPH SAMPLING

The technique we use to compress the map Z is based on a recent sampling theory for smooth signals on arbitrary graphs [12]. Let us start by recalling some concepts of graph signal processing.

3.1. Graph signal processing basics

An undirected graph $\mathcal{G}$ is a set of nodes $\mathcal{N}$, edges $\mathcal{E}$, and an adjacency matrix $\mathsf{A} \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$. The entries of A satisfy

$$\mathsf{A}[i,j] = \mathsf{A}[j,i] > 0 \text{ if } (i,j) \in \mathcal{E} \text{ and } \mathsf{A}[i,j] = 0 \text{ otherwise.} \tag{5}$$

The degree matrix $\mathsf{D} \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$ is the diagonal matrix with entries satisfying $\mathsf{D}[i,i] = \sum_{j=1}^{|\mathcal{N}|} \mathsf{A}[i,j]$. The graph Laplacian is defined as $\mathsf{L} = \mathsf{D} - \mathsf{A}$. It is a real, symmetric, positive semi-definite matrix. Its real normalised eigenvectors form an orthonormal matrix $\mathsf{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_{|\mathcal{N}|}) \in \mathbb{R}^{|\mathcal{N}|\times|\mathcal{N}|}$. The corresponding real eigenvalues are denoted $0 = \lambda_1 \leqslant \ldots \leqslant \lambda_{|\mathcal{N}|}$. The matrix U is usually viewed as the graph Fourier basis of $\mathcal{G}$ [13]. For any signal $\boldsymbol{z} \in \mathbb{R}^{|\mathcal{N}|}$ living on the nodes of $\mathcal{G}$, its Fourier representation is $\hat{\boldsymbol{z}} = \mathsf{U}^\intercal \boldsymbol{z}$. Note that the Fourier coefficients $\hat{\boldsymbol{z}}$ are ordered in increasing frequencies. A signal $\boldsymbol{z}$ is k-bandlimited on $\mathcal{G}$ if its Fourier coefficients $\hat{\boldsymbol{z}}_{k+1}, \ldots, \hat{\boldsymbol{z}}_{|\mathcal{N}|}$ are null [12, 14, 15]. More generally, we say that a signal is smooth on $\mathcal{G}$ if its energy is essentially concentrated at the lowest frequencies.

3.2. Intuition

We recall that each entry of Z indicates the active source at the corresponding TF point. Let us view Z as a signal on a graph of NF nodes – one node for each matrix entry. Imagine for a moment that the edges of this graph are such that: the nodes corresponding to source 1 are connected together and to no other nodes, the nodes corresponding to source 2 are connected together and to no other nodes, *etc*. This graph has obviously J different connected components, each one corresponding to exactly one source. Therefore, Z is constant within each component: Z is smooth on this graph. Actually, one can easily prove that Z is exactly J-bandlimited on this ideal graph and is therefore compressible. Indeed, one just needs to sample the value of one node per component to have a complete knowledge of Z. Each sample identifies the source index associated to the component and the map is reconstructed from the J samples by propagation of the sampled values to all connected nodes.

The above scenario is ideal. In practice, if we are able to construct a graph $\mathcal{G}$ such that Z is approximately k-bandlimited, then the results in [12] show that only $O(k\log(k))$ samples are sufficient to ensure a stable and accurate reconstructions of Z at the decoder. In the next sections, we detail the construction of $\mathcal{G}$ and explain the sampling and reconstruction procedures.

3.3. Graph construction

The graph $\mathcal{G}$ is used at the decoder for reconstruction and can also be used at the encoder to optimise the samples of Z to send. Therefore, we need to find a way to construct an appropriate graph $\mathcal{G}$ which is identical at the encoder and at the decoder. Note that we do not want to build the graph at the encoder and transmit it to the decoder as this would be as costly as sending Z directly. We thus construct the graph from the mixture $\boldsymbol{x}$, which is the only complete information available at both the decoder and the encoder.

NMF [11] is a well-known tool to estimate the spectral characteristics of audio signals. We propose here to use NMF to construct one feature vector per TF point, which we will use to construct $\mathcal{G}$. We first compute the power spectrogram $\mathsf{V} \in \mathbb{R}^{F\times N}$ of $\boldsymbol{x}$, $\mathsf{V}[f,n] = |\mathsf{X}[f,n]|^2$, and factorize it by solving the following optimization problem [16]

$$(\mathsf{W}^*, \mathsf{H}^*) = \underset{\mathsf{W}\in\mathbb{R}_+^{F\times Q},\, \mathsf{H}\in\mathbb{R}_+^{Q\times N}}{\operatorname{argmin}} D(\mathsf{V} \,\|\, \mathsf{WH}), \tag{6}$$

$$\text{with } D(\mathsf{V} \,\|\, \hat{\mathsf{V}}) = \sum_{f,n=1}^{F,N} d_{IS}(\mathsf{V}[f,n] \,\|\, \hat{\mathsf{V}}[f,n]),$$

where $d_{IS}(x\|y) = x/y - \log(x/y) - 1$ is the Itakura-Saito (IS) divergence. In this NMF, W^* is the spectral dictionary, H^* is the time activation matrix, and Q is the number of NMF components. To solve (6), the matrices W and H are initialized with random non-negative values and are iteratively updated via the multiplicative update rule until convergence [11, 16]. Note that the same random generator seed value can be used at the encoder and decoder to recover the same result.

When $Q \geqslant J$ is appropriately chosen, the above NMF has the tendency to isolate the spectral characteristics of each source, *i.e.*, $\mathsf{W}^*[:,l]$ is a spectral characteristic of one of the sources and $\mathsf{H}^*[l,:]$ indicates the contribution of this characteristic in the overall spectrogram at each instant. Note that one source is usually characterized by several NMF components. At each TF point (f,n), we build the following Q-dimensional feature vector $\boldsymbol{f}_{(f,n)} \in \mathbb{R}^Q$

$$\boldsymbol{f}_{(f,n)} = \Big(\mathsf{W}[f,1]\,\mathsf{H}[1,n],\ \ldots,\ \mathsf{W}[f,Q]\,\mathsf{H}[Q,n]\Big)^\intercal. \tag{7}$$

This feature vector indicates the contribution of each spectral characteristic at (f,n), showing which ones are the most active. As only one source is essentially active at each TF point, connecting feature vectors $\boldsymbol{f}_{(f,n)}$ which are similar should connect nodes for which the same source is likely to be active.

To simplify notations, let $i \in \{1, \ldots, NF\}$ index each time frequency point (f,n), and substitute $\boldsymbol{f}_i$ for $\boldsymbol{f}_{(f,n)}$. Let also $\boldsymbol{z} \in \{1, \ldots, J\}^{NF}$ be the vectorised version of Z. In order to construct $\mathcal{G}$, we connect each feature vector to its 8 nearest neighbours (in the ℓ_1 sense), which gives a set $\mathcal{E}$ of $8NF$ edges. The adjacency matrix $\mathsf{A} \in \mathbb{R}^{NF\times NF}$ of $\mathcal{G}$ satisfies $\mathsf{A}[i,i'] = 0$ for $(i,i') \notin \mathcal{E}$ and

$$\mathsf{A}[i,i'] = \exp\big[-\|\boldsymbol{f}_i - \boldsymbol{f}_{i'}\|_1/\mu\big] \text{ for } (i,i') \in \mathcal{E}, \tag{8}$$

where $\mu > 0$ is the mean of the values in the set $\{\|\boldsymbol{f}_i - \boldsymbol{f}_{i'}\|_1 : (i,i') \in \mathcal{E}\}$. We then symmetrise the matrix A and compute the graph Laplacian. As the quality of the feature vectors depends on the choice of Q, we perform several NMFs for different values of Q and concatenate all the feature vectors obtained before constructing the graph $\mathcal{G}$. This is an advantage of the proposed approach as we do not have to find the optimal value of Q as in conventional NMF based methods. Furthermore, another potential advantage is that a fusion of several NMFs with different Q may actually work better than using one single NMF, even if the choice of Q is optimised.

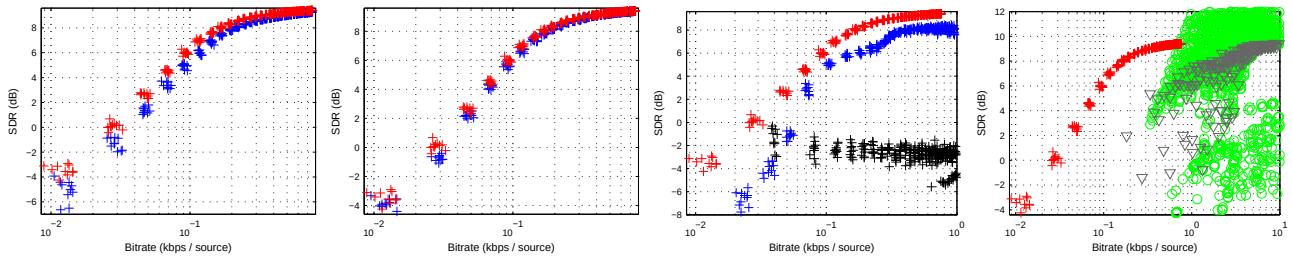

Fig. 1: SDR (dB) vs. bitrate (kbps/source). (a) Results obtained by solving (12) (red crosses) or by zero-filling (blue crosses). (b) Results obtained by concatenating feature vectors computed with $Q = 3, 6, 9$ (red crosses) or $Q = 6$ only (blue crosses). (c) Results obtained with different sampling distributions: $\boldsymbol{p}_s$ (red crosses), $\boldsymbol{p}_o$ (blue crosses), uniform distribution (black crosses). (d) Results obtained with our method (red crosses), [6] (grey triangles) and [7] (green circles).

3.4. Sampling

To sample $\boldsymbol{z}$, we follow the method described in [12]. This method relies on a sampling probability distribution defined on the nodes of $\mathcal{G}$. This distribution is represented by $\boldsymbol{p} \in \mathbb{R}^{NF}$. We obviously have $\|\boldsymbol{p}\|_1 = 1$. The i^{th} entry of $\boldsymbol{p}$, *i.e.*, $\boldsymbol{p}_i$ represents the probability of sampling node i. The samples are then chosen by selecting randomly m different nodes according to $\boldsymbol{p}$. We denote the set of selected indices $\Omega = \{\omega_1, \dots, \omega_m\} \subset \mathcal{N}$.

The results in [12] show that the efficiency of the sampling depends on the graph cumulative coherence, which is a parameter that characterises the interactions between the graph structure and the sampling distribution. In particular, it is proved in [12] that, if

$$\boldsymbol{p}_i = \|\mathsf{U}_k^\intercal \boldsymbol{\delta}_i\|_2^2 / k, \tag{9}$$

where $\mathsf{U}_k = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_k)$ and $\boldsymbol{\delta}_i$ is the Dirac at node i, then $m = O(k \log(k))$ measurements are sufficient to sample all k-bandlimited signals, with high probability. Up to the factor $\log(k)$, this is an optimal result as we necessarily need $m \geqslant k$ to sample any k-bandlimited signal. Let us highlight however that, in practical applications, if the signals of interest are not exactly bandlimited or have more intrinsic structures, the above probability distribution may not be any more the one leading to the best results. We will test different sampling distribution in our experiments.

In the following, for any vector $\boldsymbol{a} \in \mathbb{R}^{NF}$, we denote by $\boldsymbol{a}_\Omega \in \mathbb{R}^m$ its restriction to the indices in Ω. The samples sent to the decoder are thus $\boldsymbol{z}_\Omega$, after coding as described in Section 3.5. Note that we do not send the list of indices Ω at the decoder. Instead, the sampling distribution $\boldsymbol{p}$ is recomputed at the decoder and the list Ω is re-obtained at the decoder by fixing in advance the same seed value of the pseudo-random generator at the encoder and the decoder.

3.5. Differential coding

The last process in the encoder is the coding strategy used to encode the list of values in $\boldsymbol{z}_\Omega$. As only J values, or symbols, appear in $\boldsymbol{z}_\Omega$, the simplest strategy would consist in coding each value using $\log_2(J)$ bits. However, as all values do not appear with the same probability, one can achieve better results by coding this list of symbols using, *e.g.*, arithmetic coding. This is the strategy adopted here. Yet, we noticed that better results are achievable by using differential coding before arithmetic coding.

We reorder the indices in Ω as follows. We travel across the time-frequency plane starting from the lowest time index and lowest frequency, then going towards the largest time index, and continue in zigzag towards the highest frequencies. The indices are reordered in order of appearance during this travel. Even though the indices in Ω are selected at random, we noticed that for the sampling distributions that lead to the best results, this reordering makes appear sequences of constant values: the same source remains active for a while. To take advantage of this effect, we use differential coding to encode the reordered list. For simplicity, let us assume that the indices $\omega_1, \dots, \omega_m$ are ordered as just described. We compute the sequence $\hat{\boldsymbol{z}} \in \{0, \dots, J-1\}^m$ that satisfies $\hat{\boldsymbol{z}}[1] = \boldsymbol{z}[\omega_1] - 1$ and

$$\hat{\boldsymbol{z}}[i] = (\boldsymbol{z}[\omega_i] - \boldsymbol{z}[\omega_{i-1}]) \bmod J \text{ for all } i \in \{2, \dots, m\}, \tag{10}$$

which is then coded using arithmetic coding. In this work, we do not implement arithmetic coding but instead estimate the attained bitrate. We compute the probability q_j of appearance of each symbol $j \in \{0, \dots, J-1\}$. Note that we would need to transmit these J parameters for decoding in practice. We then estimate that the number of bits needed to code each symbol by arithmetic coding is $\log_2(q_j)$. The number of bits to code the sequence $\hat{\boldsymbol{z}}$ is therefore $\sum_{j=0}^{J-1} N_j \log_2(q_j)$, where N_j is the number of times j appears in the sequence.

3.6. Reconstruction

At the decoder, we have access to $\boldsymbol{z}_\Omega$ (after decoding) and we can also reconstruct the graph $\mathcal{G}$. We can thus try to estimate either $\boldsymbol{z}$ or the binary masks $\boldsymbol{m}_j^*$ – the vector $\boldsymbol{m}_j^* \in \{0,1\}^{NF}$ denotes the vectorised version of M_j^*. Indeed, if $\boldsymbol{z}$ is smooth on $\mathcal{G}$ then the masks $\boldsymbol{m}_j^*$ are also smooth on $\mathcal{G}$ by construction. Note also that the sampled binary mask $(\boldsymbol{m}_j^*)_\Omega$ can directly be deduced from Z_Ω by using Eq. 4 at the sampled TF points.

Instead of reconstructing $\boldsymbol{z}$, we choose to estimate the J masks $\boldsymbol{m}_j$ using the reconstruction method proposed in [12]. It is proved that one can stably and accurately estimate $\boldsymbol{m}_j^*$ by solving

$$\min_{\boldsymbol{m} \in \mathbb{R}^{NF}} \left\| \mathsf{P} \left[\boldsymbol{m}_\Omega - (\boldsymbol{m}_j^*)_\Omega \right] \right\|_2^2 + \gamma \, \boldsymbol{m}^\intercal \mathsf{L} \, \boldsymbol{m}, \tag{11}$$

where $\gamma > 0$ and $\mathsf{P} \in \mathbb{R}^{m \times m}$ is the diagonal matrix that satisfies $\mathsf{P}_{ii} = \boldsymbol{p}_{\omega_i}^{-1/2}$. We let the reader refer to [12] for the precise bound on the reconstruction error. In this paper, we opt for the constrained version of the above problem. We solve

$$\widetilde{\boldsymbol{m}}_j = \underset{\boldsymbol{m} \in \mathbb{R}^{NF}}{\operatorname{argmin}} \, \boldsymbol{m}^\intercal \mathsf{L} \, \boldsymbol{m} \quad \text{subject to} \quad \boldsymbol{m}_\Omega = (\boldsymbol{m}_j^*)_\Omega. \tag{12}$$

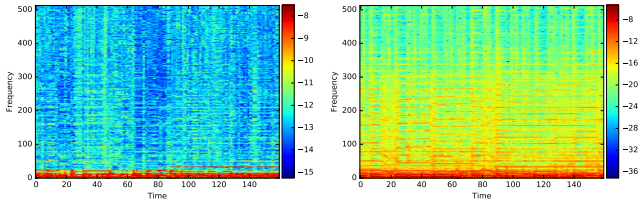

Fig. 2: Logarithm of the sampling distributions $\boldsymbol{p}_o$ obtained for $k = 0.05NF$ (left) and $\boldsymbol{p}_s$ (right).

Note that the above problem can be solved efficiently using, *e.g.*, FISTA [17], which only involves one matrix-vector multiplication with the sparse Laplacian matrix L at each iteration. The sources are then separated from the mixture using the estimated masks $\widetilde{\boldsymbol{m}}_j$.

4. EXPERIMENTS

We conduct experiments using $J = 3$ sources of a music recording of 5.5 seconds sampled at 16 kHz. For computational reasons when constructing the graph $\mathcal{G}$, we divide these recordings into segments of 1024 ms[2] and we compress them independently with our method. The STFT is computed using a half-overlapping Hann window of 1024 samples. The performance is evaluated by the signal-to-distortion ratio (SDR) after separation of the 3 sources from the mixture. The SDR is a benchmarked metric measured in dB grading the overall signal distortion [18]. The reported results are obtained by averaging the SDRs obtained after compression and separation for the 3 sources. The bitrate is estimated as described in Section 3.5.

With our method, the number of measurements m varies between 5 and 15 percent of NF in all experiments. We perform 10 simulations per number of measurements corresponding to 10 independent draws of Ω. Each cross in the plots of Fig. 1 corresponds to one experimental result.

4.1. Quality of the signal model

We first want to confirm that considering that the binary masks are smooth on $\mathcal{G}$ is a valid signal model for reconstruction. Therefore, we compare the source separation quality when the binary masks are obtained by solving (12) or by zero-filling, which consists in keeping the known value of $\boldsymbol{m}_j^*$ in Ω and setting to 0 all the values outside of Ω. In both cases, the graph is obtained by computing and concatenating the feature vectors with $Q = 3, 6, 9$. The sampling distribution $\boldsymbol{p}$ is obtained by normalising the power spectrogram V of the mixture. We denote this distribution $\boldsymbol{p}_s$. This sampling distribution favours the selection of measurements where the sources have most of their energy. The results are presented in Fig. 1a.

We notice an advantage of using the graph regularisation at low bitrate with an improvement of around 1 dB on average. At high bitrate, the performance of both methods saturates to reach an SDR of about 9.40 dB. Note that the best result one can achieve with a complete knowledge of the ideal binary masks $\boldsymbol{m}_j^*$ is 9.55 dB. We are thus close to this SDR at the highest bitrate tested, *i.e.*, for $m = 0.15NF$. There is thus less room for improvement when using the graph at high bitrate. This result also indicates that most of the information about the binary masks is contained in the 15 percent of measurements selected.

[2]Mirroring is used for the last segment which is shorter than 1024 ms.

4.2. Influence of the number of NMF decompositions

We study now the effect of concatenating or not the feature vectors obtained by NMF for different values of Q. We run experiments where the graph is obtained by computing feature vectors with $Q = 6$ only or by concatenating those obtained at $Q = 3, 6, 9$. The sampling distribution used is $\boldsymbol{p}_s$. The binary masks are recovered by solving (12). The results are presented in Fig. 1b.

On average, we notice a slight improvement of using multiple NMFs at different values of Q to construct the graph. We also observed the same behaviour when the graph is constructed with $Q = 3$ or 9 only. The graph is thus slightly better estimated when concatenating the results of different NMFs.

4.3. Influence of the sampling distribution

In this third set of experiments, we study the effect of different sampling distributions on the attained separation quality. We test the uniform sampling distribution, the distribution $\boldsymbol{p}_s$, and the distribution defined in (9) (denoted $\boldsymbol{p}_o$). This last distribution is estimated using the fast algorithm presented in [12]. The parameter k in (9) is adapted to m. We use $k = m/3$ in this set of experiments. The graph is obtained by computing and concatenating the feature vectors obtained with $Q = 3, 6, 9$. The binary masks are recovered by solving (12). The results are presented in Fig. 1c.

We observe that the uniform distribution yields the worse results. The quality does not even increase with the bit rate. The distributions $\boldsymbol{p}_s$ and $\boldsymbol{p}_o$ yield the best results, with the first one performing the best. For illustration, we show in Fig. 2 the logarithm of these distributions. Both distributions concentrate most of the measurements at low frequencies but $\boldsymbol{p}_o$ allows the highest frequencies to be sampled with higher probability than $\boldsymbol{p}_s$. While $\boldsymbol{p}_o$ depends on the structure of $\mathcal{G}$ only, $\boldsymbol{p}_s$ prevents to place measurements where none of the sources has energy. We expect that the best distributions are obtained, *e.g.*, from a mixture of these two distributions so as to obtain a distribution adapted to both the graph $\mathcal{G}$ and the energy distribution of the sources. We however have not yet explored this possibility, which is left as future work.

4.4. Comparison to other methods

Finally, we compare our method with the ISS methods presented in [6] and [7]. The graph is obtained by concatenating the feature vectors obtained at $Q = 3, 6, 9$. We use the sampling distribution $\boldsymbol{p}_s$. The binary masks are recovered by solving (12). The results are presented in Fig. 1d.

We clearly see that the advantage of the proposed method at low bitrate where it outperforms [6] and [7]. At high bitrate, while the performance of our method saturates, the method presented in [7] allows to attain a better separation quality.

5. CONCLUSION

We propose a novel ISS method for audio coding based on recent sampling developments for smooth signals on graphs. We show that this method achieves better separation quality at the decoding side for low bitrate than other state-of-the-art methods. To improve even further the method, it would be interesting to find how to compute a sampling distribution optimised for both the graph structure and the specific energy distribution of the mixture in the time-frequency plane. Note also that NMF is just one possibility to construct $\mathcal{G}$. Other methods, such as, *e.g.*, in [19] might improve the results.

6. REFERENCES

[1] E. Vincent, S. Araki, F. Theis, G. Nolte, P. Bofill, H. Sawada, A. Ozerov, V. Gowreesunker, D. Lutter, and N. Q. K. Duong, "The Signal Separation Campaign (2007-2010): Achievements and remaining challenges," *Signal Processing*, vol. 92, pp. 1928–1936, 2012.

[2] E. Vincent, N. Bertin, R. Gribonval, and F. Bimbot, "From blind to guided audio source separation: How models and side information can improve the separation of sound," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 107–115, May 2014.

[3] J. Fritsch and M.D. Plumbley, "Score informed audio source separation using constrained nonnegative matrix factorization and score synthesis," in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 888–891.

[4] L. Le Magoarou, A. Ozerov, and N. Q. K. Duong, "Text-informed audio source separation. example-based approach using non-negative matrix partial co-factorization," *Journal of Signal Processing Systems*, pp. 1–5, 2014.

[5] M. Parvaix and L. Girin, "Informed source separation of linear instantaneous under-determined audio mixtures by source index embedding," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1721–1733, 2011.

[6] A. Liutkus, J. Pinel, R. Badeau, L. Girin, and G. Richard, "Informed source separation through spectrogram coding and data embedding," *Signal Processing*, vol. 92, no. 8, pp. 1937–1949, 2012.

[7] A. Ozerov, A. Liutkus, R. Badeau, and G. Richard, "Coding based informed source separation: Nonnegative tensor factorization approach," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1699–1712, August 2013.

[8] J. Engdegard, B. Resch, C. Falchand O. Hellmuth, J. Hilpert, A. Holzer, L. Terentiev, J. Breebaart, J. Koppens, E. Schuijers, and W. Oomen, "Spatial audio object coding (SAOC) - The upcoming mpeg standard on parametric object based audio coding," in *124th Audio Engineering Society Convention (AES)*, May 2008.

[9] Çağdaş Bilen, Alexey Ozerov, and Patrick Pérez, "Compressive sampling-based informed source separation," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on*. IEEE, 2015, pp. 1–5.

[10] Ö. Yılmaz and S. T. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural and Information Processing Systems 13*, 2001, pp. 556–562.

[12] G Puy, N. Tremblay, R. Gribonval, and P. Vandergheynst, "Random sampling of bandlimited signals on graphs," *Appl. Comput. Harmon. Anal., in press,*, 2016.

[13] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *Signal Processing Magazine, IEEE*, vol. 30, no. 3, pp. 83–98, May 2013.

[14] S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, "Discrete signal processing on graphs: Sampling theory," *Signal Processing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2015.

[15] A. Anis, A. Gadde, and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 3864–3868.

[16] C. Févotte, N. Bertin, and J. Durrieu, "Non-negative matrix factorization with the Itakura-Saito divergence. with application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, 2009.

[17] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[18] E. Vincent, R. Gribonval, and C. Fvotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[19] F. Bach and M. Jordan, "Learning spectral clustering, with application to speech separation," *J. Mach. Learn. Res.*, vol. 7, pp. 1963–2001, 2006.