Fusion methods for speech enhancement and audio source separation

Xabier Jaureguiberry, Emmanuel Vincent, Gael Richard

Fusion methods for speech enhancement and audio source separation

Xabier Jaureguiberry, Emmanuel Vincent and Gaël Richard

***Abstract*—A wide variety of audio source separation techniques exist and can already tackle many challenging industrial issues. However, in contrast with other application domains, fusion principles were rarely investigated in audio source separation despite their demonstrated potential in classification tasks. In this paper, we propose a general fusion framework which takes advantage of the diversity of existing separation techniques in order to improve separation quality. We obtain new source estimates by summing the individual estimates given by different separation techniques weighted by a set of fusion coefficients. We investigate three alternative fusion methods which are based on standard non-linear optimization, Bayesian model averaging or deep neural networks. Experiments conducted for both speech enhancement and singing voice extraction demonstrate that all the proposed methods outperform traditional model selection. The use of deep neural networks for the estimation of time-varying coefficients notably leads to large quality improvements, up to 3 dB in terms of signal-to-distortion ratio (SDR) compared to model selection.**

***Index Terms*—audio source separation, fusion, aggregation, ensemble, deep neural networks, deep learning, variational Bayes, model averaging, non-negative matrix factorization, speech enhancement, singing voice extraction**

I. Introduction

Blind audio source separation aims at recovering the audio signals, called *sources*, that compose a given mixture. The most challenging situation is at stake when the number of sources is greater than the number of observable channels in the mixture. As such, the problem becomes underdetermined. Numerous approaches have been proposed in the literature [1], [2]. The sources can be modeled based on their sparsity [3], [4], their redundancy [5], [6], their spatial diversity [7], their morphological characteristics [8]–[10] or according to perceptual grouping criteria [11]. Amongst the existing source models, Non-negative Matrix Factorization (NMF) is one of the most popular [12]–[14]. For example, it has achieved great performance in the latest CHiME contest [14], [15] dedicated to speech enhancement and has also been successfully implemented for musical source separation [9]. More recently, the progress made in training deep neural networks (DNNs) has been exploited in order to estimate time-frequency masks [16] or magnitude source spectrograms [17].

Faced with a given source separation problem to be solved, one will typically either develop his/her own approach or choose an existing technique and adapt it to the problem at play. This choice is guided by the type of mixture and sources to be separated and it often leads to a compromise between separation quality and complexity of implementation. Once a technique has been chosen, the quality of separation also depends on the tuning of its parameters which is often driven by experience. For instance, the order of an NMF model is known to have a great influence on separation quality [18]. Automatic tuning based on model-order selection principles derived from information theory [19] or specific selection criteria [20], [21] might be applied but with limited success on real data [22]. Furthermore, since two distinct separation methods may have complementary strengths and weaknesses, selecting one method rather than another is expected to be suboptimal.

Fusion techniques [23], also named ensemble or aggregation techniques, precisely aim at combining several methods in order to better solve a given problem. Transposed to the context of source separation, fusion is opposed to selection as it consists in using several separation methods and combining their solutions rather than selecting the best solution according to some criterion. Fusion has been particularly popular in classification [24] and has led to efficient concepts such as bagging and boosting [25]. Despite being similar to a classification problem [26], audio source separation has barely benefited from fusion principles so far. Recently, the concept of bagging for convolutive blind source separation was introduced in [27] while the authors in [26], [28] proposed to combine time-frequency masks in a way similar to classification.

Following our previous works [22], [26], [29], we here propose a general framework for fusion in audio source separation which only assumes that the considered separation techniques lead to time-domain estimated signals. This allows the combination of heterogeneous separation techniques as well as identical techniques with different parameter settings.

In this study, we further extend our preliminary works [26], [29] to a novel adaptive time-varying fusion rule in which the fusion weights are adapted to each frame of the mixture to be separated. As such, the general fusion framework that we propose nicely handles all the previously introduced fusion rules. The objective thus turns out to be the estimation of either time-invariant or time-varying fusion coefficients, in a static or adaptive way. Note that although it is here presented for single channel signals, our fusion framework can be easily extended to the multichannel case.

Compared to our previous works, we also introduce improved learning methods for the time-invariant fusion case. For time-varying fusion, we propose two distinct approaches. The first one is based on variational Bayesian (VB) averaging and the second one aims at learning time-varying fusion coefficients with a neural network. Two source separation tasks have been retained for experimental evaluation of our fusion methods : a speech enhancement task and a singing voice extraction task. For speech enhancement, the fusion of NMFs and DNNs is studied. We particularly focus our analysis on

This work was partly supported under the research programme EDiSon3D (ANR-13-CORD-0008-01) funded by ANR, the French State agency for research.

the fusion of NMFs of different orders so as to compare all the proposed methods, including VB averaging. Experiments on the musical task further demonstrate that our fusion framework can be used in different source separation contexts.

The structure of the rest of the paper is as follows. The general framework is introduced in Section II. The estimation of time-invariant static fusion coefficients is investigated in Section III. A VB algorithm for the estimation of adaptive time-invariant and time-varying fusion coefficients is presented in Section IV. In Section V, we exploit neural networks to determine time-varying adaptive fusion coefficients. The proposed fusion rules are compared in Section VI in the context of speech enhancement using both NMF-based and DNN-based separators while the fusion of heterogeneous separation techniques is studied in Section VII. Finally, conclusions are drawn in Section VIII.

II. General fusion framework

A. Single-channel source separation

Throughout this paper, we will consider the source separation problem which consists in estimating the J sources $s_j(t)$ that compose an observable linear mixture $x(t)$. The mixing equation can be written in the Short-Time Fourier Transform (STFT) domain as

$$x_{fn} = \sum_{j=1}^{J} s_{j,fn} + \epsilon_{fn} \tag{1}$$

in which f and n respectively denote the frequency bin and the time frame. In the following, we refer to $\mathbf{s}_{fn} = [s_{1,fn} ..., s_{J,fn}]^\intercal$ as the source vector and to ϵ_{fn} as the sensor noise. Note that, here, we will not aim at dereverberating the source signals, which is sometimes one of the source separation objectives. Reverberation will not be neglected but considered as part of the source signals $s_{j,fn}$ instead.

B. Fusion of different source estimates

Let us suppose that M distinct models and/or algorithms can be used to estimate each of the J sources. We define a new estimate of each source through a simple weighted sum of the M estimated sources $\tilde{s}_{jm,fn}$ indexed by model m:

$$\forall j, f, n,\ \tilde{s}_{j,fn} = \sum_{m=1}^{M} \alpha_{m,fn}\, \tilde{s}_{jm,fn}, \tag{2}$$

in which $\forall m, f, n, \alpha_{m,fn} \geq 0$ and $\sum_{m=1}^{M} \alpha_{m,fn} = 1$. In the following, we refer to $\boldsymbol{\alpha}_{fn} = \{\alpha_{m,fn}\}_{m=1..M}$ as the set of *fusion coefficients*, or simply the *fusion vector*.

C. Time-invariant vs. time-varying fusion

Several special cases of the above general fusion rule can be considered. One such special case, called *time-invariant fusion*, is to assume that the fusion coefficients α_m remain independent of the time-frequency bin (f, n). This assumption leads to a simplified expression of (2) which turns out to be equivalent to a weighted sum of the estimated time-domain source signals:

$$\forall j, t,\ \tilde{s}_j(t) = \sum_{m=1}^{M} \alpha_m \tilde{s}_{jm}(t). \tag{3}$$

To go further, we propose in this work to investigate another special case of the time-frequency fusion rule (2) in which the fusion coefficients $\alpha_{m,n}$ depend on time only:

$$\forall j, f, n,\ \tilde{s}_{j,fn} = \sum_{m=1}^{M} \alpha_{m,n}\, \tilde{s}_{jm,fn}. \tag{4}$$

Similarly to above, this fusion rule, called herein *time-varying fusion*, can be rewritten in the time domain. Denoting $\tilde{s}^n_{jm}(t)$ the m^{th} estimation of source j within time frame n, that is the inverse STFT of $\{\tilde{s}_{jm,fn}\}_{f=1..F}$, the resulting framed source signal is expressed as

$$\tilde{s}^n_j(t) = \sum_{m=1}^{M} \alpha_{m,n} \tilde{s}^n_{jm}(t). \tag{5}$$

Contrary to (3), the fusion coefficients now depend on the frame n. The full estimated source $\tilde{s}_j(t)$ is then recovered by summing $\tilde{s}^n_j(t)$ over n in a traditional overlap-add manner.

D. Static vs. adaptive fusion

Two distinct study cases can already been derived from (3) and (5) according to whether the fusion coefficients α_m depend on the observed mixture $x(t)$ or not. In the following, we will refer to *static fusion* when the fusion coefficients do not depend on the mixture (see Section III) and to *adaptive fusion* when the fusion coefficients are estimated according to the mixture to be separated (see Sections IV and V).

E. Oracle estimation

In audio source separation, the quality of separation is often measured by the Signal-to-Distortion Ratio (SDR) expressed in decibels (dB) [30]. For instance, the SDR of the source estimate $\tilde{s}_j(t)$ is given by

$$\text{SDR}\left[\tilde{s}_j\right] = 10\log_{10} \frac{\sum_t \|s_j(t)\|^2}{\sum_t \|s_j(t) - \tilde{s}_j(t)\|^2} \tag{6}$$

where $s_j(t)$ denotes the true source signal.

As the true sources are not available in practice, fusion results which rely on their knowledge will be called *oracle* [31], so as to emphasize that they do not account for achievable results in practical situations but that they give instead an upper bound on the performance that can be expected from a given fusion rule.

As such, for a given mixture $x(t)$, we define the oracle time-invariant fusion coefficients as the coefficients α_m that maximize the SDR of the estimated source $\tilde{s}_j(t)$. They are obtained by solving the following maximization problem under linear equality and inequality constraints:

$$\begin{aligned} &\underset{\{\alpha_m\}_{m=1..M}}{\text{argmax}} \ 10\log_{10} \frac{\sum_t \|s_j(t)\|^2}{\sum_t \left\|s_j(t) - \sum_{m=1}^{M} \alpha_m \tilde{s}_{jm}(t)\right\|^2} \\ &\text{subject to} \quad \begin{cases} \forall m,\ \alpha_m \geq 0 \\ \sum_{m=1}^{M} \alpha_m = 1 \end{cases}. \end{aligned} \tag{7}$$

This turns out to be equivalent to the minimization of the source mean square error (SMSE), *i.e.*, the mean square error (MSE) between the true source $s_j(t)$ and its fused estimate

$\sum_{m=1}^{M} \alpha_m \tilde{s}_{jm}(t)$, which can be formulated as a standard Quadratic Programming (QP) [32] problem in matrix form

$$\begin{array}{ll} \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} & c_j + \boldsymbol{\alpha}^\intercal \widetilde{\mathbf{G}}_j \boldsymbol{\alpha} - 2\, \tilde{\mathbf{d}}_j^\intercal \boldsymbol{\alpha} \\ \text{subject to} & \left\{ \begin{array}{l} \forall m,\, \alpha_m \geq 0 \\ \sum_{m=1}^{M} \alpha_m = 1 \end{array} \right. \end{array} \tag{8}$$

in which $\boldsymbol{\alpha}$ denotes the vector of fusion coefficients and $\boldsymbol{\alpha}^\intercal$ its transpose. The matrix $\widetilde{\mathbf{G}}_j$ of size $M \times M$ is the so-called Gram matrix whose elements are the scalar products between the estimated signals, *i.e.*,

$$\forall m_1, m_2,\, \widetilde{G}_{j,m_1 m_2} = \sum_t \tilde{s}_{jm_1}(t) \tilde{s}_{jm_2}(t). \tag{9}$$

Similarly, the vector $\tilde{\mathbf{d}}_j$ of length M is composed of the scalar products between the estimated signals and the true source signal and c_j is the squared norm of the true source signal:

$$\begin{array}{c} \forall m, \quad \tilde{d}_{j,m} = \sum_t s_j(t) \tilde{s}_{jm}(t) \\ c_j = \sum_t \left\| s_j(t) \right\|^2 . \end{array} \tag{10}$$

Oracle results for time-varying fusion can be similarly computed by replacing the estimated and the true sources by their framed versions $\tilde{s}_{jm}^n(t)$ and $s_j^n(t)$. The components $\widetilde{\mathbf{G}}_j$, $\tilde{\mathbf{d}}_j$ and c_j are thus to be computed on each frame n and not on the whole signal.

III. Static fusion

Assuming that we have defined a subset of M separation systems that are relevant for a given source separation problem, static fusion aims at estimating a unique vector of fusion coefficients for the whole signal, each coefficient being independent of the mixture $x(t)$ to be separated. In this context, the time-invariant rule (3) and the time-varying rule (5) are strictly equivalent. In this section, we propose three distinct methods to estimate static fusion coefficients.

A. Static fusion by mean

The first, simplest method consists in taking the mean of the M estimated signals $\tilde{s}_{jm}(t)$, which is equivalent to setting $\forall m, \alpha_m = 1/M$ in (3). In the following, we will refer to this approach as *static fusion by mean*.

B. Learned static fusion

As an alternative, we propose a learning method to determine the static fusion coefficients from a representative training dataset. To do so, we proposed in [26] to solve a QP problem similar to (8) which was equivalent to minimizing the MSE between the true and estimated sources on the training dataset. Supposing that our training dataset is composed of L mixtures $x^{(l)}(t)$ together with their true sources $s_j^{(l)}(t)$, we thus wish to solve the following minimization problem

$$\begin{array}{ll} \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} & \sum_l c_{j,l} + \boldsymbol{\alpha}^\intercal \left(\sum_l \widetilde{\mathbf{G}}_{j,l} \right) \boldsymbol{\alpha} - 2 \left(\sum_l \tilde{\mathbf{d}}_{j,l}^\intercal \right) \boldsymbol{\alpha} \\ \text{subject to} & \left\{ \begin{array}{l} \forall m,\, \alpha_m \geq 0 \\ \sum_{m=1}^{M} \alpha_m = 1 \end{array} \right. , \end{array} \tag{11}$$

in which $\widetilde{\mathbf{G}}_{j,l}$, $\tilde{\mathbf{d}}_{j,l}$ and $c_{j,l}$ are defined as in (9) and (10) but for each example l. In the following, the coefficients thus obtained will be referred to as *MSE-based static fusion* coefficients.

Here, to go further, we propose to optimize the coefficients α_m in order to maximize the average SDR on the training dataset. This turns out to be equivalent to solving the following minimization problem :

$$\begin{array}{ll} \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} & \sum_l 10 \log_{10} \left(c_{j,l} + \boldsymbol{\alpha}^\intercal \widetilde{\mathbf{G}}_{j,l} \boldsymbol{\alpha} - 2\, \tilde{\mathbf{d}}_{j,l}^\intercal \boldsymbol{\alpha} \right) \\ \text{subject to} & \left\{ \begin{array}{l} \forall m,\, \alpha_m \geq 0 \\ \sum_{m=1}^{M} \alpha_m = 1 \end{array} \right. . \end{array} \tag{12}$$

In the following, the fusion coefficients thus obtained will be called *SDR-based static fusion* coefficients.

Both MSE-based and SDR-based static coefficients α_m can be used to separate any other mixture $x(t)$ which is not present in the training dataset. Note that alternative choices for the objective function could also be considered such as the Signal-to-Interference Ratio (SIR), the Signal-to-Artifacts Ratio (SAR) or a combination of these measures [30].

IV. Adaptive fusion using variational Bayesian averaging

The quality of separation can theoretically be improved by adapting the fusion coefficients to the mixture to be separated. In this context, Bayesian model averaging [33] is the reference approach to combine several estimates of a given distribution. In this section, we propose to introduce this principle in an audio source separation context and to show how it fits in our general fusion framework.

A. Bayesian model averaging principle

In the Bayesian paradigm, the sources to be estimated are usually represented as random variables that we will here symbolically denote as $\mathbf{S}_j$. The objective of source separation thus consists in estimating the posterior probability of the sources $\mathbf{S}_j$ given some observations $\mathbf{X}$ and the probabilistic model at stake $\mathcal{M}_m$. Supposing that each source $\mathbf{S}_j$ is represented by $F \times N$ independent random variables $s_{j,fn}$ in the time-frequency domain, its posterior probability can be factored as $p(\mathbf{S}_j|\mathbf{X}, \mathcal{M}_m) = \prod_{fn} p(s_{j,fn}|\mathbf{X}, \mathcal{M}_m)$. Given now that M models $\mathcal{M}_m$ are used to estimate M such posterior probabilities, the computation of the posterior probability of each model $p(\mathcal{M}_m|\mathbf{X})$ allows us to select the best estimation $p(\mathbf{S}_j|\mathbf{X}, \mathcal{M}_{m^*})$ (or equivalently the posteriors $p(s_{j,fn}|\mathbf{X}, \mathcal{M}_m)$) according to

$$m^* = \underset{m}{\operatorname{argmax}} \quad p(\mathcal{M}_m|\mathbf{X}). \tag{13}$$

To go further, Bayesian model averaging proposes to average the M estimated posterior probabilities, each being weighted by the posterior probability of the corresponding model. As such, a new posterior probability can be formulated as

$$p(s_{j,fn}|\mathbf{X}) = \sum_{m=1}^{M} p(\mathcal{M}_m|\mathbf{X})\, p(s_{j,fn}|\mathbf{X}, \mathcal{M}_m). \tag{14}$$

Thanks to Bayes rule, the posterior probability $p(\mathcal{M}_m|\mathbf{X})$ of a model can be obtained as

$$p(\mathcal{M}_m|\mathbf{X}) = \frac{p(\mathcal{M}_m)\, p(\mathbf{X}|\mathcal{M}_m)}{\sum_{m'=1}^{M} p(\mathcal{M}_{m'})\, p(\mathbf{X}|\mathcal{M}_{m'})} \tag{15}$$

where $p(\mathbf{X}|\mathcal{M}_m)$ is the *marginal likelihood* of model $\mathcal{M}_m$. By denoting $\pi_m = p(\mathcal{M}_m)$, $\tilde{s}_{j,fn} = \mathbb{E}[p(s_{j,fn}|\mathbf{X})]$ and $\tilde{s}_{jm,fn} = \mathbb{E}[p(s_{j,fn}|\mathbf{X}, \mathcal{M}_m)]$, we can then notice that (14) turns out to be equivalent to the general fusion rule (2) with

$$\alpha_m \propto \pi_m \, p(\mathbf{X}|\mathcal{M}_m). \tag{16}$$

In a Bayesian framework, the estimation of time-invariant fusion coefficients as defined in (3) is thus equivalent to the estimation of the posterior probability of each model. As such, the proposed fusion rule becomes adaptive.

B. Adaptive fusion as Bayesian model averaging

It is well known that Bayesian inference is often intractable in practice and that approximate inference is thus required. Amongst approximate methods, Variational Bayesian (VB) inference [34] is of particular interest for our study. Indeed, VB inference gives an approximation of the log-marginal likelihood of a model through the so-called *free-energy* denoted as $\mathcal{L}_m$ for model $\mathcal{M}_m$ (see Appendices A and B). The adaptive Bayesian fusion coefficients can thus be approximated as

$$\alpha_m \propto \pi_m \exp^{\mathcal{L}_m}. \tag{17}$$

C. Controlling the shape of the model posterior

In practice, preliminary experiments conducted on the fusion of NMF models of different order (see Section VI for details) have shown that the fusion coefficients estimated according to (17) practically result in a selection instead of a fusion, *i.e.*, one fusion coefficient is equal to 1 and the others are equal to 0. Moreover, it turns out that the selected model is not always the one which gives the best separation quality. As a consequence, we propose to introduce a parameter $\beta \geq 1$ that aims at controlling the shape of the posterior $p(\mathcal{M}_m|\mathbf{X})$ by penalizing its entropy [35]. The fusion coefficients are thus given by

$$\alpha_m \propto \pi_m \exp^{\mathcal{L}_m/\beta}. \tag{18}$$

D. Learning the priors and the shape parameter

The prior probabilities π_m as well as the shape parameter β must be learned in order to make (18) practicable. Similarly to the learned static fusion approaches introduced in Section III-B, we propose to learn them by optimizing either the MSE or the SDR on a representative training dataset. With $\boldsymbol{\alpha}_l(\boldsymbol{\pi}, \beta)$ denoting the set of fusion coefficients for example l which depends on the vector of prior probabilities $\boldsymbol{\pi} = (\pi_1, ..., \pi_M)^\intercal$, the shape parameter β and the vector of free energies $\boldsymbol{\mathcal{L}}_l = (\mathcal{L}_1^{(l)}, ..., \mathcal{L}_M^{(l)})^\intercal$, the optimization problem can be written in the MSE case as

$$\begin{aligned} \underset{\boldsymbol{\pi},\beta}{\operatorname{argmin}} \quad & \textstyle\sum_l \Big(c_{j,l} - 2\, \tilde{\mathbf{d}}_{j,l}^\intercal \, \boldsymbol{\alpha}_l(\boldsymbol{\pi}, \beta) \\ & \qquad + \boldsymbol{\alpha}_l(\boldsymbol{\pi}, \beta)^\intercal \, \widetilde{\mathbf{G}}_{j,l} \, \boldsymbol{\alpha}_l(\boldsymbol{\pi}, \beta) \Big) \\ \text{subject to} \quad & \forall m, \pi_m \geq 0 \end{aligned} \tag{19}$$

or in the SDR case as

$$\begin{aligned} \underset{\boldsymbol{\pi},\beta}{\operatorname{argmin}} \quad & \textstyle\sum_l 10 \log_{10} \Big(c_{j,l} - 2\, \tilde{\mathbf{d}}_{j,l}^\intercal \, \boldsymbol{\alpha}_l(\boldsymbol{\pi}, \beta) \\ & \qquad + \boldsymbol{\alpha}_l(\boldsymbol{\pi}, \beta)^\intercal \, \widetilde{\mathbf{G}}_{j,l} \, \boldsymbol{\alpha}_l(\boldsymbol{\pi}, \beta) \Big) \\ \text{subject to} \quad & \forall m, \pi_m \geq 0 \end{aligned} \tag{20}$$

Due to the introduction of the shape parameter β, solving (19) and (20) on a given training dataset is much more complex than for static fusion as the related optimization problems become non-linear under non-linear constraints. However, trust region algorithms [36] may reach satisfactory local minima.

E. Extension to time-varying fusion

The estimation of adaptive time-invariant fusion coefficients that we have derived so far can also be extended to the estimation of time-varying fusion coefficients. Indeed, as detailed in Appendix B, the free energy $\mathcal{L}_{m,n}$ can be estimated for each frame n of the observed mixture. Time-varying fusion coefficients are then formulated for each frame n as

$$\alpha_{m,n} \propto \pi_m \exp^{\mathcal{L}_{m,n}/\beta}. \tag{21}$$

This time, the priors π_m and the shape parameter β can be learned on a representative dataset by replacing the summation on l in (19) and (20) by a summation on both l and n. Moreover, $\widetilde{\mathbf{G}}_{j,l}$, $\tilde{\mathbf{d}}_{j,l}$ and $c_{j,l}$ must be replaced by their framed counterparts $\widetilde{\mathbf{G}}_{j,l,n}$, $\tilde{\mathbf{d}}_{j,l,n}$ and $c_{j,l,n}$.

V. Adaptive fusion using neural networks

The adaptive fusion scheme presented in Section IV requires a Bayesian treatment of source separation which may not be available for other models than NMF. Moreover, as it will be demonstrated in Section VI, adaptive time-varying fusion as presented in Section IV-E does not improve the results compared to time-invariant fusion, while oracle time-varying fusion exhibits a great potential. As a consequence, we propose in this section to resort to (potentially deep) neural networks in order to determine time-varying fusion coefficients and get closer to oracle performance.

A. Problem formulation

Given a representative training dataset composed of several mixtures with the corresponding true and estimated sources, we wish to estimate the fusion coefficients from the knowledge of the mixture and the estimated sources only. Traditionally, such an estimation is conducted in two steps. The first step consists in computing some features of the inputs, namely the mixture signal $x(t)$ and the M estimated signals $\tilde{s}_{jm}(t)$. The second step aims at mapping these features to the desired output, here the oracle vector of fusion coefficients $\{\alpha_{m,n}\}$.

For the feature extraction step, the set of potential features is extremely large and the selection of an appropriate subset varies with respect to the type of mixture and sources at play. For instance, we can name Mel-Frequency Cepstrum Coefficients (MFCCs), Linear Prediction Coefficients (LPC), chroma and so on. For the mapping step, Gaussian Mixture Models (GMMs) have been widely used, notably in speech recognition.

However, DNNs have now outperformed GMMs in many fields [37]. The main advantage of DNNs is their ability to perform the feature extraction step and the mapping step jointly. Furthermore, the recent introduction of *rectified linear units* (ReLUs) [38] replacing traditional hyperbolic tangent and logistic activations has brought even more advantages such as faster and more accurate convergence. Neural networks thus seem to be quite promising in our case.

B. Network architecture

Some considerations about the architecture of the neural network required for such a task can already be discussed, without any experimental context. Concerning the dimensionality of the problem, we can expect to have a small output layer of size M (*i.e.*, one neuron per fusion coefficient) but a much larger input layer. In the following, we will consider that the input is composed of the short-term power spectra of the available signals.

More precisely, in our experiments detailed in Sections VI and VII, the input relative to frame n is defined as follows:

$$\begin{bmatrix} |\mathbf{x}_{n-C}|^2 & \cdots & |\mathbf{x}_n|^2 & \cdots & |\mathbf{x}_{n+C}|^2 \\ |\tilde{\mathbf{s}}_{j1,n-C}|^2 & \cdots & |\tilde{\mathbf{s}}_{j1,n}|^2 & \cdots & |\tilde{\mathbf{s}}_{j1,n+C}|^2 \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ |\tilde{\mathbf{s}}_{jm,n-C}|^2 & \cdots & |\tilde{\mathbf{s}}_{jm,n}|^2 & \cdots & |\tilde{\mathbf{s}}_{jm,n+C}|^2 \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ |\tilde{\mathbf{s}}_{jM,n-C}|^2 & \cdots & |\tilde{\mathbf{s}}_{jM,n}|^2 & \cdots & |\tilde{\mathbf{s}}_{jM,n+C}|^2 \end{bmatrix} \tag{22}$$

in which the first line refers to the mixture with $\mathbf{x}_n = [x_{1n}, ..., x_{fn}, ..., x_{Fn}]$ and the next M lines refer to the M estimated sources with $\tilde{\mathbf{s}}_{jm,n} = [\tilde{s}_{jm,1n}, ..., \tilde{s}_{jm,fn}, ..., \tilde{s}_{jm,Fn}]$. Each line is composed of the power spectra of the current frame, the C preceding frames and the C following frames. We hence take advantage of the context of the central frame n. Each frame being a vector of F frequency bins, the final input which results from the flattening of matrix (22) into one dimension is a vector of length $F(2C+1)(M+1)$.

In order to decrease the size of the neural network, a reduction of the input dimensionality is desirable. For this purpose, Singular Value Decomposition (SVD) or Principal Component Analysis (PCA) are both commonly used to reduce high-dimensional data while retaining relevant components of the input [34]. For our experiments, we chose to use PCA so as to keep a certain amount of data variance. Moreover, the training data have been standardized, *i.e.*, centered and normalized to unit variance, before and after PCA.

The output layer, which corresponds to the M estimated time-varying fusion coefficients $\tilde{\alpha}_{m,n}$, is composed of M neurons, one for each coefficient. A *softmax* activation function is used to ensure that $\forall n, \sum_m \tilde{\alpha}_{m,n} = 1$. Other layers are made of ReLUs.

C. Training cost functions

A neural network is usually trained by gradient descent together with backpropagation of errors with respect to a given cost function to be minimized [39]. The cost function is often defined as a function of the estimated output, here the estimated coefficients $\{\tilde{\alpha}_{m,n}\}$ for time frame n, and of the desired output, here the oracle coefficients $\{\alpha_{m,n}\}$ as defined in Section II-E. A common choice is to minimize the mean square error. For a given frame n, the MSE is defined as

$$\varphi_n^{\text{OMSE}} = \sum_{m=1}^{M} \left(\alpha_{m,n} - \tilde{\alpha}_{m,n}\right)^2. \tag{23}$$

In the following, we will refer to this cost as the Oracle MSE (OMSE). Another common choice is to use the Cross-Entropy (CE) generalized to non-binary multiclass problems which is defined in our context as

$$\varphi_n^{\text{CE}} = -\sum_{m=1}^{M} \alpha_{m,n} \log \frac{\tilde{\alpha}_{m,n}}{\alpha_{m,n}}. \tag{24}$$

Note that both these cost functions require the oracle fusion coefficients to be known in order to estimate the errors.

Hereafter, we propose two other cost functions that do not require the knowledge of oracle fusion coefficients. Following the SMSE optimization formulations of (11) and (19), the cost function for training can be defined as the MSE between the n^{th} frame of the true source $s_j^n(t)$ and of its estimate $\tilde{s}_j^n(t)$ obtained from the corresponding estimated fusion coefficients, *i.e.*, the outputs of the network $\tilde{\alpha}_{m,n}$. For frame n, the cost function is defined as

$$\varphi_n^{\text{SMSE}} = c_{j,n} + \tilde{\boldsymbol{\alpha}}_n^\intercal \widetilde{\mathbf{G}}_{j,n} \tilde{\boldsymbol{\alpha}}_n - 2\, \tilde{\mathbf{d}}_{j,n}^\intercal \tilde{\boldsymbol{\alpha}}_n \tag{25}$$

where $c_{j,n}$, $\tilde{\mathbf{d}}_{j,n}$ and $\widetilde{\mathbf{G}}_{j,n}$ are defined in (9) and (10). Following (12) and (20), the cost function can also be defined as the SDR of the n^{th} frame of source estimate $\tilde{s}_j^n(t)$ obtained from the estimated fusion coefficients $\tilde{\alpha}_{m,n}$. The cost function thus becomes :

$$\varphi_n^{\text{SDR}} = 10 \log_{10} \left(c_{j,n} + \tilde{\boldsymbol{\alpha}}_n^\intercal \widetilde{\mathbf{G}}_{j,n} \tilde{\boldsymbol{\alpha}}_n - 2\, \tilde{\mathbf{d}}_{j,n}^\intercal \tilde{\boldsymbol{\alpha}}_n \right). \tag{26}$$

Note that in order to make the training as efficient as with the two other cost functions, $c_{j,n}$, $\widetilde{\mathbf{G}}_{j,n}$ and $\tilde{\mathbf{d}}_{j,n}$ can be precomputed for all frames n of the training dataset.

Finally, we shall remark that neural networks are usually trained by iterative *mini-batch* gradient descent so that each iteration aims at minimizing the mean cost function over a small sample $\mathcal{B}$ of the training dataset, called a *mini-batch*. The total cost function to be minimized at each iteration is thus

$$\varphi = \frac{1}{|\mathcal{B}|} \sum_{n \in \mathcal{B}} \varphi_n \tag{27}$$

in which $|\mathcal{B}|$ is the mini-batch size. As a consequence, in an on-line setting, *i.e.*, when $|\mathcal{B}| = 1$, the source MSE cost (25) and the SDR cost (26) are strictly equivalent. In the following however, the mini-batch size has been fixed to $|\mathcal{B}| = 50$. In this case, note that the SDR cost (26) computed on a mini-batch is different from the SDR of this mini-batch.

D. Other settings

At each iteration, the frames which compose a mini-batch are randomly picked from the training dataset. At the end of each epoch (*i.e.*, when all training frames have been presented exactly once), the performance of the current network is evaluated on a validation dataset. According to the average validation score, the learning rate is adapted for next epoch and *early stopping* may be performed following the method proposed in [40]. The training is stopped either when the validation score does not evolve anymore or when a predefined maximum number of epochs is attained. For networks with more than one hidden layer, the discriminative pre-training proposed in [41] has been used. The performance of the final network is then evaluated on a test dataset. For our experiments, the neural networks have been implemented thanks to the Python library *Theano* [42], [43].

VI. Experimental evaluation on a speech enhancement task

In this section, we propose to evaluate and compare all the fusion techniques described in Sections III, IV and V on a speech enhancement scenario. As Bayesian model averaging introduced in Section IV requires to employ probabilistic models only, we will first restrain our study to the fusion of NMF-based separators. Afterwards, we will study the adaptive fusion of both NMF-based and DNN-based separators.

A. CHiME corpus

For this experiment, we rely on the second CHiME challenge corpus [44]. The signals are composed of speech utterances from 34 distinct speakers overlapped with noise signals recorded in a real domestic environment.

We divided the data into four disjoint datasets :

- a *clean training dataset* which features 500 utterances in clean conditions (*i.e.*, reverberated but without background noise) for each speaker,
- a *training dataset* composed of 600 utterances, each mixed with background noise at six different SNRs,
- a *validation dataset* composed of 300 utterances, each mixed with background noise at six different SNRs,
- and a *test dataset* also composed of 300 utterances, each mixed with background noise at six different SNRs.

The background noise has been randomly chosen in order to reach the six different SNRs, namely $\{-6$ dB, -3 dB, 0 dB, 3 dB, 6 dB, 9 dB$\}$. The clean training dataset has been used to learn speaker-dependent NMF models of different orders (see Section VI-B). These clean utterances have also been mixed at six different SNRs with background noise so as to train the DNN-based separators in Section VI-H6. The training dataset has been used to learn static fusion coefficients (see Section III), to learn the priors π_m and the shape parameter β (see Section IV) and to train the neural networks (see Section IV). The validation set has been used in the fusion scheme based on neural network in order to perform *early stopping* and learning rate adaptation according to [40]. Finally, the test set aims at evaluating and comparing all proposed separation and fusion techniques. Note that the test set has never been used either to learn the fusion coefficients or to optimize the architecture of neural networks.

B. NMF-based separators

Single-channel speech enhancement aims at cleaning up a speech signal s_1 from a background noise s_2. Both the speech and the noise signals are supposed to follow the mixing equation (1). In order to compare all the proposed fusion methods, the sources will be here modelled by NMF either by using the standard maximum-likelihood formulation of NMF (ML-NMF) presented in Appendix A or its variational Bayesian formulation (VB-NMF) presented in Appendix B.

The separation performance obtained with NMF is known to depend on the number of components, denoted as K_j, chosen to model each source [18], [26]. Few works [20], [21] have implemented model selection principles in order to infer the best number of components for a given problem. As an alternative to these selection approaches, we here investigate the use of fusion to combine several NMF models of different orders.

As the background noise signal s_2 exhibits less variability than the speech signal s_1, we will here keep the number of components of the background model to a constant value, *i.e.*, $K_2 = 32$ and will only vary the number of components of the speech model K_1. Precisely, we will consider $M = 7$ possible numbers of components $K_{1m} = 2^m$ with $m = 1..M$. For each mixture, the speaker is supposed to be known. For each number of components K_{1m}, the separation process is then conducted in two stages. At first, an NMF of K_{1m} components is estimated on the concatenation of all speaker utterances available in the clean training dataset. The dictionary $\mathbf{W}_{1m}$ thus learned forms a spectral model of K_{1m} components for this speaker. In parallel, a spectral model $\mathbf{W}_2$ of the background noise is learned by NMF on the available noise excerpts that precede and follow the utterance in the mixture. In both cases, the dictionaries have been initialized by vector quantization and the corresponding activation matrices have been initialized randomly. In a second step, both the speech NMF model and the background NMF model are re-estimated on the mixture so as to estimate the speech signal s_1 by traditional Wiener filtering [45]. To do so, the dictionaries $\mathbf{W}_{1m}$ and $\mathbf{W}_2$ are fixed to the spectral models learned at the previous step. Their corresponding activation matrices are initialized with the averaged values estimated at the previous step as well. In both steps, the estimation process has been stopped after 50 iterations. In the end, by repeating this procedure for each K_{1m}, we have M different estimates $\tilde{s}_{1m}$ of the speech source, based on different spectral models $\mathbf{W}_{1m}$ that all describe the same speaker but with different resolutions. For instance, the model with $K_{11} = 2$ components gives a very rough description of the speaker spectral characteristics whereas the model with $K_{17} = 128$ components will give a much more detailed description but with potential redundancies.

The time-frequency representation used in this experiment is the Quadratic Equivalent Rectangular Bandwidth (QERB) transform [46] with half-overlapping sine windows of 1024 samples and $F = 350$ frequency bins. Indeed, the ERB frequency scale (which is similar to the Mel scale) has been shown to result in better separation performance than the STFT [47]. This same representation has been used for both the separation step and the computation of the neural network inputs.

C. Evaluation measure

As mentioned above, the SDR is the most used evaluation measure in audio source separation. Methods will thus be compared with respect to the SDR of the estimated speech signals. The performance on a given utterance is measured as the global SDR on the whole utterance as defined in (6). It is then averaged over all utterances in the test dataset. We have also systematically measured the quality of separation through the computation of the average MSE. Nevertheless, since the MSE appeared to follow the same trends as the SDR, we will only report the latter hereafter.

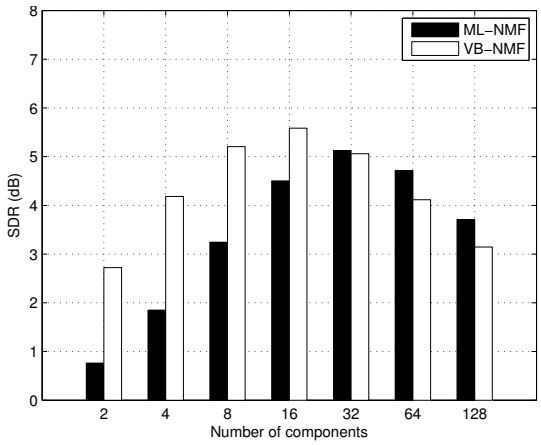


Fig. 1. Separation performance of ML-NMF and VB-NMF as a function of the number of components.

D. ML vs. VB inference for NMF

As a baseline, the SDRs of the speech signals estimated with both ML-NMF and VB-NMF are drawn in Fig. 1, as a function of the number of components M . The highest performance is obtained using VB-NMF with 16 components (5.59 dB on average). The graph also suggests that VB-NMF better accommodates few components whereas ML-NMF performs better for many components. However, as it requires more parameters to be estimated, VB-NMF is less computationally efficient than ML-NMF (14.5 seconds vs. 12.8 seconds on average per utterance).

E. Oracle fusion

Oracle fusion requires the knowledge of the true sources that compose each mixture. As such, it gives an upper bound of the fusion performance that we can expect in practice. Fig. 2 depicts the results of oracle fusion in both time-invariant and time-varying cases. The times frames for time-varying fusion have been computed with half-overlapping sine windows of 1024 samples. Oracle fusion results are to be compared with *oracle selection* results which are also drawn in Fig. 2 and which consist in selecting for each utterance the best performing model in terms of SDR instead of combining the M models as in fusion. As expected, fusion outperforms selection in all considered situations. The oracle time-invariant fusion of ML-NMFs brings a gain of 0.7 dB SDR over oracle time-invariant selection, while for VB-NMF, the gain is of 0.4 dB. Moreover, oracle time-varying fusion allows a gain of nearly 3 dB SDR compared to oracle time-invariant fusion.

F. Static fusion

Contrary to oracle fusion, static fusion as introduced in Section III does not require the knowledge of the true sources. MSE-based and SDR-based static fusion results depicted in Fig. 3 thus account for practical results in which the fusion coefficients are learned on the training dataset by solving the minimization problems (11) and (12) respectively. As in the oracle case, we compare these results with *SDR-based static selection* which consists in retaining the individual model that performs best on the training dataset in terms of SDR. As an upper limit, the performance of oracle time-invariant fusion is also recalled for both ML-NMF and VB-NMF.

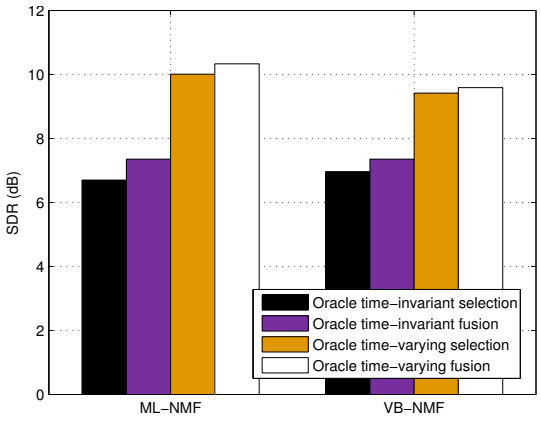


Fig. 2. Performance of oracle time-invariant and time-varying fusions compared to oracle time-invariant and time-varying selections, for both ML-NMF and VB-NMF.

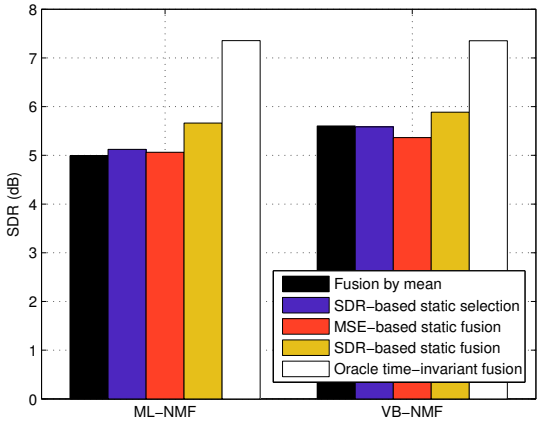


Fig. 3. Performance of static time-invariant fusion compared to static time-invariant selection, for both ML-NMF and VB-NMF.

While MSE-based static fusion does not improve separation quality compared to SDR-based static selection, SDR-based static fusion allows us to improve upon selection by 0.6 dB with ML-NMF and 0.3 dB with VB-NMF. These results again demonstrate the interest of fusion over selection.

Furthermore, it is worth noting that simple fusion by mean gives interesting performance. Indeed, contrary to the others, this fusion approach does not need any training and performs similarly to SDR-based static selection.

G. Adaptive fusion using variational Bayesian averaging

The performance of static fusion methods are satisfactory compared to traditional selection techniques but the adaptation of the fusion coefficients to the mixture at play could allow us to get closer to oracle performance. Fig. 4 depicts the results obtained by adaptive time-invariant and time-varying fusion using variational Bayesian averaging, as presented in Section IV, for both MSE (19) and SDR (20) optimization. Similarly to oracle and static fusion, we compared fusion results with their corresponding selection approaches. Here, VB selection consists in applying the Bayesian selection criterion (13), *i.e.*, retaining for each utterance the model whose free energy is maximum. As an upper limit, the performance of oracle time-invariant and time-varying fusions are also recalled.

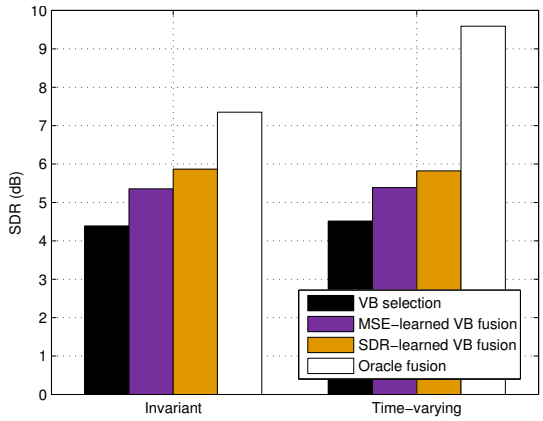


Fig. 4. Performance of adaptive time-invariant and time-varying fusions using VB-NMF, for both MSE and SDR optimization.

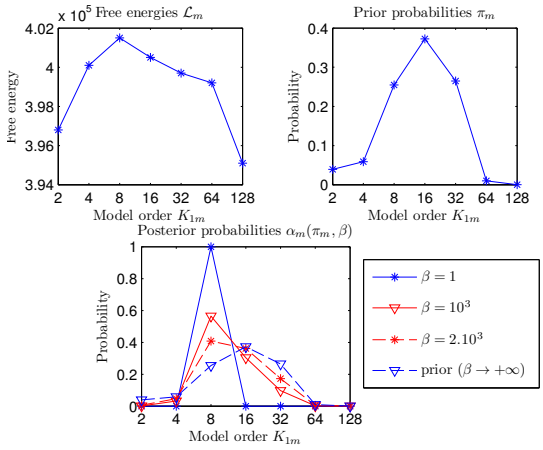


Fig. 5. Shape of the order posterior obtained by VB fusion as a function of the shape parameter β, for one utterance of the CHiME corpus.

We recall that VB selection is similar to VB fusion with $\beta = 1$. To illustrate this, we have drawn in Fig. 5 the values of the free energies $\mathcal{L}_m$, the learned prior probabilities π_m and the resulting posterior probabilities (18) for different values of β. For $\beta = 1$, it is clear that the posterior probability is equal to 1 for $K_{1m} = 16$, *i.e.*, where the free energy is maximum. When $\beta \to +\infty$, the posterior is equal to the prior. Thus, a value $\beta \in [1, +\infty[$ allows us to make a compromise between the prior and the VB criterion.

The results of Fig. 4 demonstrate that the introduction of the shape parameter β effectively allows us to outperform VB selection. As for static fusion, we can notice that the learning of the parameters with respect to the SDR is more efficient than with respect to the MSE. In both the invariant and the time-varying case, the gain in SDR is of 0.5 dB. Compared to VB selection, the gain reaches up to 1.5 dB.

However, the overall performance of VB fusion is mixed when compared with static fusion in Fig. 3. Indeed, VB selection is largely outperformed by SDR-based static selection and fusion by mean by 1 dB SDR on average, which demonstrates that the VB criterion does not correlate well with separation quality. While the introduction of the shape parameter β partly compensates for this defect, it turns out that both time-invariant VB fusion and time-varying VB fusion perform similarly to SDR-based static fusion. Two main reasons might be responsible for these mixed results. At first, we shall recall that the objective functions (19) and (20) that we wish to minimize to learn the priors and the shape parameter are more complex to optimize than the objective functions (11) and (12) at play in the static case. Despite this, further experiments have shown that the found minima are global. Therefore, these mixed results are most probably due to the approximate inference strategy that is required in the VB-NMF scheme and which affects the quality of the Bayesian criterion. While it used to be the reference approach in the literature [48], [49], our experiments thus show that the VB criterion is ineffective to selecting the best number of components in terms of separation quality.

H. Adaptive time-varying fusion using neural networks

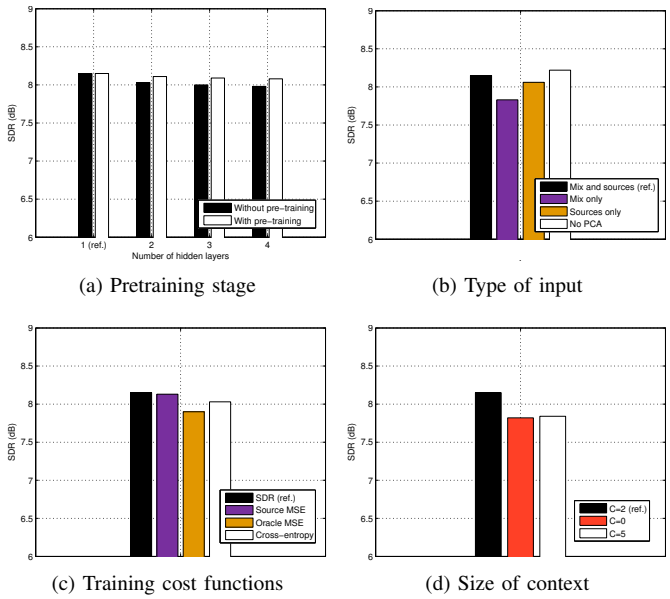
Adaptive time-varying fusion based on variational Bayesian averaging has failed to improve separation performance compared to its time-invariant counterpart. Yet, oracle time-varying fusion results show that time-varying fusion has a great potential. As we will show in this subsection, the adaptive time-varying fusion framework based on neural networks, presented in Section V, allows us to get closer to oracle performance.

The search for the best neural network architecture for our problem has been conducted through the testing of several architectures, from single-layer networks to deeper networks. Notably, the number of hidden layers has been varied from one to four layers and the number of units per layer has been defined as multiples of the output layer size M. We have tested 11 layer sizes, namely $\{7, 14, 28, 56, 112, 224, 448, 896, 1792, 3584, 7168\}$. We here considered the ML-NMF formulation as the resulting oracle time-varying performance is greater than for VB-NMF. These architectures will be reviewed in the next subsections. To start with, let us introduce the architecture with a unique hidden layer that performs best.

1) Best single-layer architecture: The best architecture has been selected according to its performance on the validation dataset. The input was composed as defined in Sections V-B and VI-B. We chose a context of size $C = 2$. We computed a PCA on the whole input in order to keep 85 % of the variance, which allows us to reduce the input dimensionality from 14000 to 154. The data have also been standardized, i.e., centered and normalized to unit variance, before and after PCA. The best results on the validation set have been obtained using the SDR cost as defined in Section V-C with a single-layer network composed of 1792 hidden units. As one might expect, the performance monotonically increases with the number of hidden units from 7 to 1792 hidden units and decreases afterward.

This best single-layer network finally leads to an average SDR of 8.15 dB on the test set. Adaptive fusion using neural networks thus outperforms all aforementioned fusion techniques and allows a gain of 3 dB SDR compared to SDR-based static selection. It even outperforms oracle time-invariant fusion by almost 0.9 dB.

2) Influence of the architecture: As mentioned above, deeper architectures have been investigated. We have varied the number of hidden layers from 2 to 4. In each case, the



(a) Pretraining stage (b) Type of input

(c) Training cost functions (d) Size of context

Fig. 6. Influence on the test SDR of other architecture parameters.

number of neurons has been fixed for all hidden layers to the same value amongst the set of 11 different values given above. However, none of these architectures outperformed the best single-layer network described earlier.

The influence of the pre-training phase has also been investigated. The results obtained with and without pre-training are drawn in Fig. 6a for networks with 1792 neurons per layer. We can notice that the increase of the number of hidden layers tends to slightly degrade the performance of fusion. These results also show that the pre-training phase allows a gain of 0.1 dB SDR on average.

Though it can be surprising with respect to the recent progress brought by deep learning in speech recognition, we think that this behaviour is explained by the type of problem we want to solve. The same behaviour has been observed in [50] in the context of multimedia event detection. The authors suggest that single-layer networks might outperform deeper ones for learning tasks which require a high level of abstraction, which is typically the case in our situation. Moreover, in a source separation context, recent works such as [16] have shown that single-layer networks already perform well and that the improvement brought by deeper networks is marginal.

3) Influence of the training cost functions: The results obtained by varying the cost function used for training have been drawn in Fig. 6c. It is shown that both the SDR (26) and SMSE (25) cost functions which are related to the separation objective bring a gain of 0.2 dB SDR compared to the more standard cost functions (23) and (24).

4) Influence of the type of input: The influence of the choice of the input is depicted in Fig. 6b. While the first bar refers to our reference architecture with takes the full matrix (22) as an input to the PCA, the second bar refers to a neural network whose input only takes into account the mixture, *i.e.*, the very first line of matrix (22). Similarly, the third bar refers to a network whose input only takes into account the estimated sources, *i.e.*, the M last lines of matrix (22). Both these results are outperformed by our best architecture by 0.3 dB and 0.1 dB respectively. The very last bar of this graph shows that keeping all the variance of the input, *i.e.*, not reducing the dimensionality of the full matrix (22) by PCA, brings an improvement of 0.07 dB. Note that this tiny improvement is obtained at the expense of a much longer training time (3 hours and 45 minutes without PCA against 38 minutes with PCA on a 64-bit Linux machine with an *NVIDIA Quadro 600* GPU and a quad-core *Intel Xeon* CPU).

Training cost function	Number of hidden layers	Test SDR (dB)
MSE	1	8.13
MSE	2	8.83
MSE	3	8.85
SA	1	8.32
SA	2	8.28
SA	3	8.53
MSE+SA	1	8.34
MSE+SA	2	8.98
MSE+SA	3	9.01

TABLE I
SDR (DB) ACHIEVED BY DNN-BASED MASK ESTIMATION.

5) Influence of the context size: Finally, Fig. 6d illustrates the influence of the context size. It shows that a network in which no context is taken into account ($C = 0$) leads to a drop of 0.35 dB SDR in comparison with our baseline architecture whose context size is $C = 2$. More surprisingly, a larger context size of $C = 5$ also gives worse performance than our baseline.

6) Fusion of heterogeneous methods: Until now, we have studied the fusion of NMF-based separators only so as to compare all our fusion methods, including VB averaging. Recently, DNNs have been applied to the direct estimation of time-frequency masks [16], [17]. Such DNN-based separators now seem to outperform NMF-based ones. As such, we propose in this section to compare their separation performance and to study if fusion can still help improving separation quality.

For this experiment, we have retained the feed-forward solution proposed by [16]. We have trained 9 such DNNs that vary either by the cost function that has been used for training (MSE, SA or MSE+SA, see [16] for details) or by the number of hidden layers (from 1 to 3). The DNNs have been trained on a different training set than for the fusion step, *i.e.*, the original *training set* of the CHiME corpus [44]. Separation results are reported in Table I. The best separation performance reaches 9.01 dB SDR, which is indeed far beyond the best results obtained by NMF-based separation.

We have then studied the fusion of these DNN-based separators. Oracle and practical results are reported in Table II. The first three lines recall the results obtained in the case of the fusion of NMFs only. The performance of the best NMF separator is recalled so as to highlight the improvement brought by fusion. The 95 % confidence intervals are also given to assess the significance of given SDRs and of relative SDR improvements. The three following lines refer to the fusion of the 9 DNN-based separators. As for the fusion of NMF, we have tested several architectures and we report here the one that gave the best performance on the validation dataset. The so-called best architecture is a single-layer network of 2304 hidden units which has been optimized with respect to the SDR cost function and whose input is obtained by PCA of the full matrix (22) with a context of

Sep.	Fusion approach	SDR (dB)	SDRI (dB)
NMFs	Best individual separator	5.12 ± 0.20	–
	Adaptive time-varying fusion	8.15 ± 0.16	3.03 ± 0.13
	Oracle time-varying fusion	10.33 ± 0.15	5.21 ± 0.13
DNNs	Best individual separator	9.01 ± 0.17	–
	Adaptive time-varying fusion	9.30 ± 0.17	0.30 ± 0.03
	Oracle time-varying fusion	10.26 ± 0.15	1.25 ± 0.15
DNNs + NMFs	Best individual separator	9.01 ± 0.17	–
	Adaptive time-varying fusion	9.50 ± 0.16	0.49 ± 0.06
	Oracle time-varying fusion	11.71 ± 0.14	2.70 ± 0.15

TABLE II
TEST SDRS AND SDR IMPROVEMENTS (SDRIS) IN DB AND WITH 95 % CONFIDENCE INTERVALS FOR ORACLE SELECTION AND ORACLE AND ADAPTIVE TIME-VARYING FUSIONS OF DIFFERENT SETS OF SEPARATORS (SEP. IN FIRST COLUMN) : NMFS ONLY, DNNS ONLY OR NMFS AND DNNS. THE SDRI IS COMPUTED WITH RESPECT TO THE BEST INDIVIDUAL SEPARATOR.

$C = 2$. It leads to a gain of 0.3 dB SDR in comparison with the best DNN-based separator. Note that this improvement is statistically significant. Moreover, oracle time-varying fusion results that are also reported in Table II show a potential room for improvement of 1 dB. Both these results interestingly demonstrate that adaptive time-varying fusion of homogeneous separators can still help improving separation quality when considering other types of separators than the previously studied NMFs.

Results for the heterogeneous fusion of both NMF-based and DNN-based separators are finally reported in the last lines of Table II. The potential given by the oracle time-varying fusion performance is about 1.5 dB greater than for the fusion of NMFs only and the fusion of DNNs only, which demonstrates that heterogeneous fusion can take advantage of the diversity of the separators to be combined. In practice, adaptive time-varying fusion finally reaches 9.50 dB SDR on the test set, which accounts for an improvement of 0.49 dB with respect to the best DNN-based separator.

7) Diversity of separators: As expected, our study tends to show that fusion takes advantage of the diversity of the separators. While diversity might be easily obtained by considering different types of separators as in heterogeneous fusion, the results in Table II suggest that considering a unique kind of separator and varying its parameters as in homogeneous fusion can also lead to satisfying diversity and interesting fusion performance. Indeed, the individual performance of NMF-based separators as depicted in Fig. 1 features a lot of variability in the measured SDRs. As a consequence, oracle time-varying fusion shows a potential gain of more than 5 dB compared to the best individual separator. On the contrary, the individual performance of DNN-based separators as reported in Table I is far less variable, which seems to lower the potential of time-varying fusion down to 1.3 dB only. As such, both heterogeneous and homogeneous fusion are of interest. Finally, it is important to note that, in some situations, homogeneous fusion might be the only practicable scheme as some specific source separation problems cannot be solved with several separators. Moreover, implementing several methods for one source separation problem might be too time consuming, while homogeneous fusion might feature enough diversity to reach interesting fusion performance.

VII. EXPERIMENTAL EVALUATION ON MUSIC

In this section, we propose to evaluate our approach for the fusion of heterogeneous separation methods on a singing voice extraction task. The goal is to separate the main voice signal from its musical accompaniment.

A. Music dataset

In [6], a musical dataset has been gathered from the community music remixing website *ccMixter*[1]. It features 49 full-length stereo tracks from diverse musical genres. For our experiments, the tracks have been randomly divided into 5 groups of similar size in order to evaluate our fusion techniques by cross-validation. Furthermore, each of the 49 tracks has been cut into non-overlapping chunks of length comprised between 20 and 30 seconds, which results in a total of 308 excerpts.

For learning purposes, three of the five groups form the learning dataset whereas the two remaining groups respectively account for the validation and test sets. In order to evaluate our fusion techniques on all tracks, all experiments have been repeated five times so that each group of tracks has been used as a validation test and a test set once, following the principle of cross-validation.

B. Separation and fusion techniques

Each excerpt has been processed with four different separation techniques, namely : the Instantaneous Mixture Model (IMM) proposed in [51], Robust Principal Component Analysis (RPCA) presented in [52], the Repeating Pattern Extraction Technique based on similarity (REPETsim) of [53] and the Kernel Additive Model (KAM) described in [6].

We here propose to compare static time-invariant fusion introduced in Section III to adaptive time-varying fusion using neural networks introduced in Section V, as the considered separation techniques do not fit in a common probabilistic framework in order to apply adaptive Bayesian fusion as presented in Section IV.

C. Results

All results are gathered in Table III. They are given in terms of SDR and averaged across all excerpts of each group and, in the last column, across all excerpts of the database. The four first lines present the results obtained with the four considered separation methods. The IMM outperforms other methods for each group as well as on average. As a consequence, it is the separator that would have been chosen by traditional static selection. Amongst the static fusion methods presented in the next three lines, both MSE-based and SDR-based time-invariant fusion outperform the IMM separator, by respectively 0.57 dB and 0.64 dB on average.

The results of adaptive time-varying fusion using neural networks have been obtained with a single-layer network composed of 512 hidden units. As in Section VI, the input data was composed as defined in (22) with a context of size $C = 2$. We used the QERB transform as well with half-overlapping windows of 2048 samples and $F = 350$ frequency

[1] http://www.ccmixter.org

Method	Group 1	Group 2	Group 3	Group 4	Group 5	Average
IMM	3.49	4.33	3.16	2.55	2.83	3.30
RPCA	-0.92	-1.69	-3.65	-2.18	-1.22	-1.90
KAM	2.17	2.03	0.07	0.11	1.57	1.24
REPETsim	3.19	2.44	1.12	1.78	2.38	2.21
Fusion by mean	3.62	3.47	1.94	2.34	3.08	2.92
MSE-based time-invariant fusion	4.44	4.61	3.31	3.15	3.70	3.87
SDR-based time-invariant fusion	4.38	4.5	3.55	3.17	3.67	3.94
Oracle time-invariant fusion	*5.07*	*5.18*	*4.03*	*3.53*	*4.22*	*4.44*
Adaptive time-varying fusion using neural networks (SMSE)	4.92	5.01	3.81	3.35	3.91	4.20
Adaptive time-varying fusion using neural networks (SDR)	4.27	5.06	3.68	3.15	3.71	4.01
Oracle time-varying fusion	*6.88*	*7.07*	*5.89*	*5.28*	*6.08*	*6.28*

TABLE III
PERFORMANCE OF SEPARATION AND FUSION METHODS ON CCMIXTER TEST SETS.

bins. The data have been furthermore standardized and the input dimensionality has been reduced via PCA. The average separation performance is given for both the SMSE (25) and the SDR (26) cost functions.

Here again, adaptive time-varying fusion using neural networks outperforms all other proposed fusion techniques, namely fusion by mean, SDR-based time-invariant fusion and MSE-based time-invariant fusion by 1.3, 0.3 and 0.25 dB respectively. Compared to the experiments in Section VI, the gain in terms of SDR is less important, which can be explained by the task complexity. Indeed, even if the size of the training set (approximately 200,000 frames per group) is comparable to the size of the training set in the speech enhancement task, we must highlight that the ccMixter database features much more variability in its contents because it mixes music excerpts from very heterogeneous genres.

Finally, we note that, contrary to the speech enhancement experiment, the SMSE training cost function resulted in a gain of 0.2 dB compared to the SDR cost function. If this result might be surprising, we must recall that the proposed SDR cost function averages the SDRs computed on each frame of the mini-batch. This differs from computing the SDR of the mini-batch itself, which could be seen as an alternative to the so-called SMSE and SDR cost functions here studied. Nonetheless, this experiment confirms that using a training cost function related to the separation objective can help improving separation performance and suggests that the choice of the training cost function remains data-dependent in practice.

VIII. CONCLUSION

In this paper, we introduced a general fusion framework that aims at combining several source estimates. Our framework is flexible in that it makes limited assumptions on the type of separators to be fused, so that both estimates obtained from a single separation technique with different parameter settings and estimates obtained from distinct separation techniques can be considered. We presented three different ways to determine the fusion coefficients, depending on whether the fusion coefficients are adapted or not to the signal to be separated. Moreover, we proposed to operate the fusion either on the whole signal or at the time frame level.

All our fusion techniques have been evaluated and compared to state-of-the-art model selection techniques on a speech enhancement task handled with NMF-based separators. In this context, fusion turned out to be always more efficient than selection. In particular, the method based on neural networks allowed us to gain 3 dB SDR compared to simple selection. These experiments also demonstrated that, while Bayesian inference is the reference approach in model selection, it did eventually not outperform the simpler static fusion rule.

Additional experiments have shown that variability plays an important role in fusion. The study of the fusion of both DNN-based and NMF-based separators demonstrated that increased variability in the source estimates to be fused can boost the potential of fusion. However, the experiments conducted on a singing voice extraction task suggested that such variability is not sufficient to reach satisfying practical fusion performance and that the training dataset has to be representative enough of the learning task at stake. Indeed, an insightful look at the values of the oracle time-varying fusion coefficients highlights that their distribution is much more spread in the music scenario than in the speech one, which makes the learning task more complex. As such, we believe that the success of adaptive time-varying fusion based on neural networks depends on both the variability of the estimates and the quality of the training dataset. To confirm this, we will consider in future works other source separation tasks as well as some alternatives to neural networks such as GMMs.

To extend this work, we will also study other objective functions for the determination of fusion coefficients. These functions could be any measure relevant to a given source separation problem, such as the SIR, the SAR or even a combination of these measures. We also plan to extend our fusion framework to the handling of frequency-varying and time-frequency-varying fusion rules.

APPENDIX
NON-NEGATIVE MATRIX FACTORIZATION

A. Maximum-likelihood formulation

When using NMF for audio source separation [45], it is usually assumed that the Power Spectral Density (PSD) $v_{j,fn}$ of each source at time n and frequency f can be modeled as the result of NMF [54]

$$v_{j,fn} = \sum_{k=1}^{K_j} w_{j,fk} h_{j,kn}, \tag{28}$$

in which the coefficients $\{w_{j,fk}\}_{f=1..F}^{k=1..K_j}$ form the *dictionary* of spectral templates characterizing the spectral content of the j^{th} source and $\{h_{j,kn}\}_{k=1..K_j}^{n=1..N}$ are the so-called *activation*

coefficients which indicate the amplitude of activation of each spectral template across time. K_j is the *number of components*, *i.e.*, the number of spectral templates chosen to model source j.

Parameter estimation is performed by iterative multiplicative update rules which minimize a certain divergence d between the mixture PSD $|x_{fn}|^2$ on the one hand, and the sum of the J source PSDs $v_{j,fn}$ and the noise PSD σ^2 on the other hand:

$$\underset{w,h}{\operatorname{argmin}} \sum_{fn} d\left(|x_{fn}|^2 \middle| \sum_{j=1}^{J} v_{j,fn} + \sigma^2\right). \tag{29}$$

In the following, we consider the Itakura-Saito (IS) divergence:

$$d_{IS}(x|y) = \frac{x}{y} - \log\frac{x}{y} - 1. \tag{30}$$

Once the model parameters have been estimated, the source STFT coefficients are estimated as:

$$\tilde{s}_{j,fn} = \frac{v_{j,fn}}{\sum_{j'=1}^{J} v_{j',fn} + \sigma^2}\, x_{fn}. \tag{31}$$

It can be shown that $\tilde{s}_{j,fn}$ is also the Minimum Mean Square Error (MMSE) estimate of source j [45], assuming that $s_{j,fn}$ follows a circularly-symmetric complex normal distribution $s_{j,fn} \sim \mathcal{N}(0, v_{j,fn})$.

The results of separation obtained with NMF highly depend on the number of components K_j chosen for each source [18], [26]. In a fusion context, assuming that the separation process has been conducted for M different numbers of components K_{jm} leading to M estimates $\tilde{s}_{jm}(t)$ of each source j, a new source estimate can be obtained using (2).

B. Variational Bayesian formulation

To go further, we consider the generative model of NMF introduced in [48] which allows full Bayesian treatment. Both the dictionary and the activation coefficients are seen as random variables and assumed to follow Gamma priors :

$$w_{j,fk} \sim \Gamma(a,a) \;\text{ and }\; h_{j,kn} \sim \Gamma(b,b). \tag{32}$$

The goal of Bayesian inference is to estimate the posterior probability over all model parameters. As this is generally intractable, some approximation is required. VB is a practical inference algorithm which has been successfully applied to this generative model [48], [49] in particular.

In this context, the posterior probability of the source vector $\mathbf{s}_{fn}$ is identified as a multivariate complex Gaussian distribution denoted as $q(\mathbf{s}_{fn}) = \mathcal{N}(\boldsymbol{\mu}_{\mathbf{s},fn}, \boldsymbol{\Sigma}_{\mathbf{s},fn})$. The STFT coefficients of the sources are then given by the expectation $\boldsymbol{\mu}_{\mathbf{s},fn}$ of this posterior distribution, which can be expressed in a similar form as (31) with

$$\tilde{s}_{j,fn} = \mu_{s_{j,fn}} = \frac{C_{j,fn}}{\sum_{j'=1}^{J} C_{j',fn} + \sigma^2}\, x_{fn}. \tag{33}$$

The term $C_{j,fn}$ depends on the NMF parameters of source j through the expectation

$$C_{j,fn} = \sum_{k=1}^{K_j} \mathbb{E}\left[\frac{1}{w_{j,fk}h_{j,kn}}\right]^{-1} \tag{34}$$

in which $\mathbb{E}$ denotes the expectation over the variational posteriors $q(w_{j,fk})$ and $q(h_{j,kn})$ which turn out to be generalized inverse Gaussian (GIG) distributions. For the detailed expressions of $\boldsymbol{\Sigma}_{\mathbf{s},fn}$, $q(w_{j,fk})$, $q(h_{j,kn})$ and $C_{j,fn}$, see [29].

Practically, VB proposes to approximate the marginal likelihood of a model by the so-called *free energy*, which can then be used to select the most likely model [55], [56] amongst a set of possible models. We propose here to use the free energy to achieve fusion instead of simple selection as in [48], [57].

To that aim, we replace the log-likelihood $\log p(\mathbf{X}|\mathcal{M}_m)$ (equation (14)) with the free energy given by VB inference and denoted as $\mathcal{L}_m$:

$$\begin{aligned}\mathcal{L}_m &= \textstyle\sum_{fn} \mathbb{E}\left[\log p(x|\mathbf{s}_{fn})\right] \\ &+ \textstyle\sum_{j,fn} \left(\mathbb{E}\left[\log p(s_{j,fn}|w_j,h_j)\right] - \mathbb{E}\left[\log q(s_{j,fn})\right]\right) \\ &+ \textstyle\sum_{j,fk} \left(\mathbb{E}\left[\log p(w_{j,fk})\right] - \mathbb{E}\left[\log q(w_{j,fk})\right]\right) \\ &+ \textstyle\sum_{j,kn} \left(\mathbb{E}\left[\log p(h_{j,kn})\right] - \mathbb{E}\left[\log q(h_{j,kn})\right]\right).\end{aligned} \tag{35}$$

Here, the operator $\mathbb{E}$ denotes the expectation over the variational distributions of each parameter, *i.e.*, $q(s_{j,fn})$, $q(w_{j,fk})$ and $q(h_{j,kn})$. For details on their computation, refer to [49].

The free energy can also be computed for each frame of the signal to be separated for time-varying fusion. The free energy for frame n is then defined as

$$\begin{aligned}\mathcal{L}_{m,n} &= \textstyle\sum_{f} \mathbb{E}\left[\log p(x|\mathbf{s}_{fn})\right] \\ &+ \textstyle\sum_{j,f} \left(\mathbb{E}\left[\log p(s_{j,fn}|w_j,h_j)\right] - \mathbb{E}\left[\log q(s_{j,fn})\right]\right) \\ &+ \textstyle\sum_{j,fk} \left(\mathbb{E}\left[\log p(w_{j,fk})\right] - \mathbb{E}\left[\log q(w_{j,fk})\right]\right) \\ &+ \textstyle\sum_{j,k} \left(\mathbb{E}\left[\log p(h_{j,kn})\right] - \mathbb{E}\left[\log q(h_{j,kn})\right]\right).\end{aligned} \tag{36}$$

REFERENCES

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[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, 2014.

[3] P. D. O'Grady, B. A. Pearlmutter, and S. T. Rickard, "Survey of sparse and non-sparse methods in source separation," *International Journal of Imaging Systems and Technology*, vol. 15, no. 1, pp. 18–33, 2005.

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

[5] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[6] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet, "Kernel additive models for source separation," *IEEE Transactions on Signal Processing*, vol. 62, no. 16, pp. 4298–4310, 2014.

[7] A. Jourjine, S. Rickard, and O. Yılmaz, "Blind separation of disjoint orthogonal signals: Demixing N sources from 2 mixtures," in *Proc. of IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, vol. 5, 2000, pp. 2985–2988.

[8] D. P. W. Ellis and R. J. Weiss, "Model-based monaural source separation using a vector-quantized phase-vocoder representation," in *Proc. of IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, vol. 5, 2006, pp. V–957–960.

[9] J.-L. Durrieu, G. Richard, B. David, and C. Févotte, "Source/filter model for unsupervised main melody extraction from polyphonic audio signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 3, pp. 564–575, 2010.

[10] G. Mysore, P. Smaragdis, and B. Raj, "Non-negative hidden Markov modeling of audio with application to source separation," in *Proc. of Int. Conf. on Latent Variable Analysis and Signal Separation*. Springer, 2010, pp. 140–148.

[11] D. F. Rosenthal and H. G. Okuno, *Computational auditory scene analysis*. L. Erlbaum Associates Inc., 1998.

[12] B. Raj, R. Singh, and T. Virtanen, "Phoneme-dependent NMF for speech enhancement in monaural mixtures," in *Proc. of Interspeech*, 2011, pp. 1217–1220.

[13] N. Moritz, M. R. Schädler, K. Adiloğlu, B. T. Meyer, T. Jürgens, T. Gerkmann, B. Kollmeier, S. Doclo, and S. Goetze, "Noise robust distant automatic speech recognition utilizing NMF based source separation and auditory feature extraction," *Proc. of CHiME-2013*, pp. 1–6, 2013.

[14] J. T. Geiger, F. Weninger, A. Hurmalainen, J. F. Gemmeke, M. Wöllmer, B. Schuller, G. Rigoll, and T. Virtanen, "The TUM + TUT + KUL approach to the 2nd CHiME challenge: Multi-stream ASR exploiting BLSTM networks and sparse NMF," *Proc. of CHiME-2013*, pp. 25–30, 2013.

[15] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: An overview of challenge systems and outcomes," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 162–167.

[16] F. Weninger, J. R. Hershey, J. Le Roux, and B. Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2014, pp. 577–581.

[17] E. M. Grais, M. U. Sen, and H. Erdogan, "Deep neural networks for single channel source separation," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3734–3738.

[18] N. Bertin, R. Badeau, and G. Richard, "Blind signal decompositions for automatic transcription of polyphonic music: NMF and K-SVD on the benchmark," in *Proc. of IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, vol. 1, 2007, pp. I–65–68.

[19] A. Rabinovich, S. Belongie, T. Lange, and J. M. Buhmann, "Model order selection and cue combination for image segmentation," in *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 2006, pp. 1130–1137.

[20] V. Y. F. Tan and C. Févotte, "Automatic relevance determination in nonnegative matrix factorization with the β-divergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1592–1605, 2013.

[21] T. Virtanen, A. T. Cemgil, and S. Godsill, "Bayesian extensions to nonnegative matrix factorisation for audio signal modelling," in *Proc. of IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2008, pp. 1825–1828.

[22] X. Jaureguiberry, E. Vincent, and G. Richard, "Multiple-order nonnegative matrix factorization for speech enhancement," in *Proc. of Interspeech*, 2014, p. 4.

[23] I. Bloch, A. Hunter, A. Appriou, A. Ayoun *et al.*, "Fusion: General concepts and characteristics," *International Journal of Intelligent Systems*, vol. 16, no. 10, pp. 1107–1134, 2001.

[24] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226–239, 1998.

[25] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of online learning and an application to boosting," in *Computational learning theory*, 1995, pp. 23–37.

[26] X. Jaureguiberry, G. Richard, P. Leveau, R. Hennequin, and E. Vincent, "Introducing a simple fusion framework for audio source separation," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2013, pp. 1–6.

[27] S. Chandna and W. Wenwu, "Improving model-based convolutive blind source separation techniques via bootstrap," in *Proc. of IEEE Statistical Signal Processing Workshop (SSP)*, 2014, pp. 424–427.

[28] J. Le Roux, S. Watanabe, and J. R. Hershey, "Ensemble learning for speech enhancement," in *Proc. of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2013, pp. 1–4.

[29] X. Jaureguiberry, E. Vincent, and G. Richard, "Variational Bayesian model averaging for audio source separation," in *Proc. of IEEE Statistical Signal Processing Workshop (SSP)*, 2014, pp. 33–36.

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

[31] E. Vincent, R. Gribonval, and M. D. Plumbley, "Oracle estimators for the benchmarking of source separation algorithms," *Signal Processing*, vol. 87, no. 8, pp. 1933–1950, 2007.

[32] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[33] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky, "Bayesian model averaging: a tutorial," *Statistical science*, pp. 382–401, 1999.

[34] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[35] K. Katahira, K. Watanabe, and M. Okada, "Deterministic annealing variant of variational Bayes method," *Journal of Physics: Conference Series*, vol. 95, no. 1, 2008.

[36] T. F. Coleman and Y. Li, "An interior trust region approach for nonlinear minimization subject to bounds," *SIAM Journal on Optimization*, vol. 6, no. 2, pp. 418–445, 1996.

[37] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[38] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean *et al.*, "On rectified linear units for speech processing," in *Proc. of IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2013, pp. 3517–3521.

[39] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[40] S. Duffner and C. Garcia, "An online backpropagation algorithm with validation error-based adaptive learning rate," *Artificial Neural Networks*, pp. 249–258, 2007.

[41] D. Yu, L. Deng, F. T. B. Seide, and G. Li, "Discriminative pretraining of deep neural networks," 2011, US Patent App. 13/304,643.

[42] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: a CPU and GPU math expression compiler," in *Proc. of the Python for Scientific Computing Conference (SciPy)*, 2010.

[43] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," in *Proc. of Workshop on Deep Learning and Unsupervised Feature Learning (NIPS)*, 2012.

[44] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second 'CHiME' speech separation and recognition challenge: datasets, tasks and baselines," in *Proc. of IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2013, pp. 126–130.

[45] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[46] E. Vincent, "Musical source separation using time-frequency source priors," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 91–98, 2006.

[47] J. J. Burred and T. Sikora, "Comparison of frequency-warped representations for source separation of stereo mixtures," in *Proc. of Audio Engineering Society Convention*, 2006.

[48] M. Hoffman, D. M. Blei, and P. R. Cook, "Bayesian nonparametric matrix factorization for recorded music," in *Proc. of International Conference on Machine Learning (ICML)*, 2010, pp. 439–446.

[49] K. Adiloğlu and E. Vincent, "Variational Bayesian inference for source separation and robust feature extraction," Inria, Tech. Rep. RT-0428, 2012.

[50] M. Ravanelli, B. Elizalde, J. Bernd, and G. Friedland, "Insights into audio-based multimedia event classification with neural networks," in *Proc. of the Workshop on Community-Organized Multimodal Mining: Opportunities for Novel Solutions*, 2015, pp. 19–23.

[51] J.-L. Durrieu, B. David, and G. Richard, "A musically motivated mid-level representation for pitch estimation and musical audio source separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1180–1191, 2011.

[52] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. of IEEE International Conference on Audio, Speech and Signal Processing (ICASSP)*, 2012, pp. 57–60.

[53] Z. Rafii and B. Pardo, "Music/voice separation using the similarity matrix," in *Proc. of International Symposium on Music Information Retrieval (ISMIR)*, 2012, pp. 583–588.

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

[55] A. Corduneanu and C. M. Bishop, "Variational Bayesian model selection for mixture distributions," in *Proc. of the 8th International Workshop on Artificial Intelligence and Statistics*, 2001, pp. 27–34.

[56] J. M. Bernardo *et al.*, "The variational Bayesian EM algorithm for incomplete data: with application to scoring graphical model structures," in *Proc. of Valencia International Meeting on Bayesian Statistics*, 2002, pp. 453–462.

[57] A. T. Cemgil, "Bayesian inference for nonnegative matrix factorisation models," *Computational Intelligence and Neuroscience*, 2009, article ID 785152.

Xabier Jaureguiberry received the State Engineering degrees from Supélec (Paris, France) and from Arts et Métiers ParisTech (Paris, France) in 2010. He received the Ph.D. degree in audio signal processing from Télécom ParisTech in 2015. From 2010 to 2012, he worked as a Research Engineer at Audionamix (Paris, France) on audio source separation topics. Since 2016, he works as a Data Scientist at Zenly (Paris, France). His research interests range from audio signal processing to machine learning, with application to speech enhancement, music source separation and music information retrieval.

Emmanuel Vincent is a Research Scientist with Inria (Nancy, France). He received the Ph.D. degree in music signal processing from the Institut de Recherche et Coordination Acoustique/Musique (Paris, France) in 2004 and worked as a Research Assistant with the Centre for Digital Music at Queen Mary, University of London (United Kingdom), from 2004 to 2006. His research focuses on probabilistic machine learning for speech and audio signal processing, with application to real-world audio source localization and separation, noise-robust speech recognition, and music information retrieval. He is a founder of the series of Signal Separation Evaluation Campaigns and CHiME Speech Separation and Recognition Challenges. He was an associate editor for IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

Gaël Richard (SM'06) received the State Engineering degree from Télécom ParisTech, France (formerly ENST) in 1990, the Ph.D. degree from LIMSI-CNRS, University of Paris-XI, in 1994 in speech synthesis, and the *Habilitation à Diriger des Recherches* degree from the University of Paris XI in September 2001. After the Ph.D. degree , he spent two years at the CAIP Center, Rutgers University, Piscataway, NJ, in the Speech Processing Group of Prof. J. Flanagan, where he explored innovative approaches for speech production. From 1997 to 2001, he successively worked for Matra, Bois d'Arcy, France, and for Philips, Montrouge, France. In particular, he was the Project Manager of several large scale European projects in the field of audio and multimodal signal processing. In September 2001, he joined the Department of Signal and Image Processing, Télécom ParisTech, where he is now a Full Professor in audio signal processing and Head of the Signal and Image processing department. He is a coauthor of over 200 papers and inventor in 8 patents. He was an Associate Editor of the IEEE Transactions on Audio, Speech and Language Processing between 1997 and 2011 and one of the guest editors of the special issue on Music Signal Processing of IEEE Journal on Selected Topics in Signal Processing (2011). He currently is a member of the IEEE Audio and Acoustic Signal Processing Technical Committee, member of the EURASIP and AES and senior member of the IEEE.