Active-Set Newton Algorithm for Overcomplete Non-Negative Representations of Audio

Tuomas Virtanen*, *Member, IEEE*, Jort F. Gemmeke, *Member, IEEE*, Bhiksha Raj, *Member, IEEE*

***Abstract*—This paper proposes a computationally efficient algorithm for estimating the non-negative weights of linear combinations of the atoms of large-scale audio dictionaries, so that the generalized Kullback-Leibler divergence between an audio observation and the model is minimized. This linear model has been found useful in many audio signal processing tasks, but the existing algorithms are computationally slow when a large number of atoms is used. The proposed algorithm is based on iteratively updating a set of active atoms, with the weights updated using the Newton method and the step size estimated such that the weights remain non-negative.**

Algorithm convergence evaluations on representing audio spectra that are mixtures of two speakers show that with all the tested dictionary sizes the proposed method reaches a much lower value of the divergence than can be obtained by conventional algorithms, and is up to 8 times faster. A source separation separation evaluation revealed that when using large dictionaries, the proposed method produces a better separation separation quality in less time.

***Index Terms*—acoustic signal analysis, audio source separation, supervised source separation, non-negative matrix factorization, Newton algorithm, convex optimization, sparse coding, sparse representation**

EDICS Category: AUD-ANSY, AUD-SSEN

I. Introduction

With the rapid increase in the amount of audio data made available to public, there has been increasing need for techniques to analyze or modify the contents of these audio. In this context, *compositional* models, which characterize acoustic events through dictionaries of spectral patterns, and attempt to explain the audio as non-negative linear combinations of the dictionary atoms, are becoming increasingly popular. These models have been successfully used in many problems, such as signal analysis and recognition [1]–[3], manipulation and enhancement [3]–[7], and coding [8], [9].

The biggest advantage of the compositional model is its ability to model sound *mixtures*, which is important because real-life audio recordings typically consist of mixtures of sounds. Sounds usually mix in a *constructive* manner — when two sounds occur simultaneously their energies add in the resulting signal. The compositional model naturally captures this phenomenon by characterizing mixed signals as non-negative compositions of their component sounds.

The model characterizes the power (or magnitude) spectral vectors from each individual sound source as *constructive*, *i.e.*, a non-negative linear combinations of component spectra, which are represented as atomic entries from a dictionary. The spectrum of a mixed signal is therefore also constructive, *i.e.*, a non-negative weighted combination of the dictionary atoms from all potentially active sound sources. If properly decomposed, both the constituent sounds and their levels in the mixed signal can be determined – these are the sources underlying the dictionary atoms that have been assigned non-zero weights, and the weights themselves.

The decomposition typically attempts to minimize a divergence measure defined between the weighted combination of dictionary atoms and the power spectrum of the mixed signal. Both in *non-negative matrix factorization* (NMF) [10], [11] and *sparse representations* [12]–[14], a number of algorithms have been proposed to obtain non-negative decompositions. The choice of divergence measure affects the solution obtained. A variety of divergence measures based on the family of Bregman divergences or Csiszár divergences [11, Chapter 2], including the L_2 error [15], the Kullback-Leibler (KL) divergence [4] and the Itakura-Saito divergence [16], have been considered for optimization in the context of audio processing. For the analysis of audio data represented using magnitude or power spectra, NMF algorithms that minimize the KL divergence have been found to be particularly effective [4], [17], [18], because these spectra exhibit a large dynamic range and the KL divergence better captures the resulting non-linear relevance of the magnitudes.

The application of dictionary-based NMF solutions to generic, real-world audio has remained limited so far, because the dictionary quickly becomes extremely large as it needs to represent all the sound types that may reasonably be expected in a recording. Not only does the computational complexity increase with the size of the dictionary, conventional NMF decomposition algorithms [10], [11, pp. 267-268] require more iterations to converge as dictionary sizes increase. In this paper we propose a new *active-set* algorithm to minimize Kullback-Leibler divergences for non-negative decomposition of sound recordings using large dictionaries. The algorithm incrementally adds dictionary atoms to an active set, until a sound recording is adequately explained. The weights of the

T. Virtanen is with the Department of Signal Processing, Tampere University of Technology, Tampere, Finland e-mail: tuomas.virtanen@tut.fi. His research is funded by the Academy of Finland, grant 258708.

J. F. Gemmeke is with the Center for Processing Speech and Images, Department of Electrical Engineering of the KU Leuven, Belgium, e-mail: jgemmeke@esat.kuleuven.be. His research is funded by IWT-SBO grant 100049 (ALADIN).

B. Raj is with the Language Technologies Institute of the School of Computer Science at Carnegie Mellon University, USA e-mail: bhiksha@cs.cmu.edu

atoms in the active set are estimated using Newton's method, with the step size estimated so that the resulting weights obey the non-negativity constraints. The method is dubbed ASNA, short for Active-Set Newton Algorithm.

In contrast to previous second-order optimization algorithms [19]–[21], ASNA allows the use of overcomplete dictionaries, and in contrast to previous second-order optimization algorithms for overcomplete dictionaries [22], [23], ASNA uses a full Hessian matrix and takes advantage of the sparsity of the solution. Also, unlike previous active-set methods [24] for NMF, ASNA minimizes the KL-divergence rather than L_2 error. A good review of available algorithms from the point of view of non-overcomplete dictionaries and L_2 error is given in [25], and from the point of view of other divergences in [11]. As will be shown in this paper, overcomplete non-negative representations are sparse, which means ASNA can also be related to sparse optimization [13], [26]–[29], especially "greedy" active-set methods such as Matching Pursuit (MP) [30], Orthogonal Matching Pursuit (OMP) [31] and Compressive Sampling Matching Pursuit (CoSaMP) [32]. Still, such a relation is mainly conceptual: rather than requiring sparsity as goal or constraint to obtain feasible solutions, sparse solutions emerge naturally in ASNA as a property of the data, and we do not require any specified or minimal level of sparsity in the solution. Neither do we explicitly project our solution into sparse subspaces, although our algorithm may be viewed as a projected gradient algorithm since the step size is chosen so that weights are in the non-negative orthant.

Experimental evaluations show that ASNA is more efficient than conventional NMF solutions, particularly as the dictionary size increases. Not only does ASNA converge much faster, the experiments indicate that the conventional NMF solution does not converge within a finite number of iterations while ASNA converges in a relatively small number of iterations. As a result, the proposed algorithm may significantly enhance our ability to analyze large corpora of audio to obtain content-based descriptions.

One of the most common applications of non-negative representations is source separation, i.e., the process of estimating individual sources from a mixture [33]. We investigate the capability of ASNA to speed up source separation algorithms and find out that it results to better source separation quality vs. computation time in comparison to the baseline method.

The rest of the paper is organized as follows. In Section II we introduce the linear model, the KL-divergence and the use of non-negative, overcomplete dictionaries. In Section III we describe the proposed algorithm in detail. In Section IV we analyze properties such as sparsity, uniqueness, and computational complexity. In Section V we outline our experimental setup and in Section VI we evaluate the performance of the algorithm on representing spectra of mixtures of speech. Section VI-C evaluates the capability of ASNA to speed up existing source separation algorithms. Finally, in Section VII we present our conclusions and plans for future work.

II. Overcomplete non-negative representation of audio

A. The linear model

We operate on non-negative observation vectors $\mathbf{x}$ of length F that are for example magnitude (square root of the power) spectra of audio calculated in short frames. The observation vector $\mathbf{x}$ is modeled as a weighted linear combination of atom vectors $\mathbf{b}_n$ from a *dictionary* as follows

$$\mathbf{x} \approx \hat{\mathbf{x}} = \sum_{n=1}^{N} w_n \mathbf{b}_n, \quad \text{subject to} \quad w_n \geq 0 \ \ \forall n \tag{1}$$

where w_n, $n = 1, \ldots, N$ are the non-negative weights, n is the index of each atom, and N is the number of atoms in the dictionary. In an overcomplete representation, the number of atoms N is larger than the dimensionality F of the observations. The magnitude spectral representation of $\mathbf{x}$ permits us to interpret the atoms $\mathbf{b}_n$ as magnitude spectra of constituent sounds that superimpose to compose $\mathbf{x}$, since in a time-domain signal that is a superposition of multiple sources the magnitude spectra of the sources add approximately linearly.

By denoting the dictionary as an $F \times N$ matrix $\mathbf{B} = \begin{bmatrix}\mathbf{b}_1, \ldots, \mathbf{b}_N\end{bmatrix}$ and the weights as an $N \times 1$ weight vector $\mathbf{w} = \begin{bmatrix}w_1, \ldots, w_N\end{bmatrix}^T$, the model is rewritten as

$$\hat{\mathbf{x}} = \mathbf{B}\mathbf{w}, \quad \text{subject to} \quad \mathbf{w} \geq \mathbf{0}. \tag{2}$$

The atoms in the dictionary can be obtained e.g. by sparse NMF [34], K-SVD [35], clustering [36], [37], sampling [3], [38], or by a parametric representation [39]. The proposed method does not place any restrictions on how the atoms are obtained. We only assume that the atoms are entry-wise non-negative, given in advance, and remain fixed. Even though we use magnitude spectra as examples, the algorithm also does not assume any particular way how the observation vectors are acquired, just that they are entry-wise non-negative.

B. Quantifying the modeling error

The weights are estimated by minimizing a divergence measure between an observed vector $\mathbf{x}$ and the model $\hat{\mathbf{x}}$ in (1). It has been observed [17], [18] that measures such as the generalized Kullback-Leibler (KL) divergence [4] or Itakura-Saito (IS) [16] divergence are more appropriate for quantifying the modeling error of magnitude spectra of audio, in comparison to the Euclidean distance that is used in many other fields of science. In this work we restrict ourselves to the KL divergence, which is defined as

$$\mathrm{KL}(\mathbf{x}||\hat{\mathbf{x}}) = \sum_i d(x_i, \hat{x}_i), \tag{3}$$

where function d is defined as

$$d(p,q) = \begin{cases} p\log(p/q) - p + q & p > 0 \ \text{ and } \ q > 0 \\ q & p = 0 \\ \infty & p > 0 \ \text{ and } \ q = 0 \end{cases} \tag{4}$$

C. EM estimation

The standard method for minimizing the KL divergence under the linear model is based on initializing the weights to positive values and iteratively applying the following multiplicative update proposed by Lee and Seung in [10]:

$$\mathbf{w} \leftarrow \mathbf{w} \otimes \frac{\mathbf{B}^T(\frac{\mathbf{x}}{\mathbf{B}\mathbf{w}})}{\mathbf{B}^T\mathbf{1}}, \tag{5}$$

where $\otimes$ denotes entry-wise multiplication, divisions are done entry wise, and $\mathbf{1}$ is an all-ones vector of length F. In the rest of the paper we refer to this method as "EM", since the update rule can be derived from the expectation-maximization (EM) algorithm [40]. There exist variants of the EM algorithm that aim to improve its speed of convergence. For example the method [41] updates only one weight at the time; this is particularly effective when the dictionary is small, but is inefficient in the overcomplete case since it requires a separate update for each atom.

III. Active-set Newton algorithm for minimizing the KL divergence

The main principle of the proposed optimization method is that it estimates and updates a set of "active" atoms that have non-zero weights. The active set is initialized with a single atom in it. We then iteratively perform the following steps. We find the most promising atom not in the active set, and add it to the active set. We run several iterations of Newton's method to estimate weights for the active atoms, ensuring that all weights remain non-negative. Atoms whose weights go to zero are removed from the active set. The procedure is iterated until a convergence criterion is achieved. Each of the processing steps is explained in more detail below.

Let us denote the *active set* $\mathcal{A}$ as the set of indices of dictionary atoms with non-zero weights. The model $\hat{\mathbf{x}}$ for the observation vector at each iteration of the algorithm is written as

$$\hat{\mathbf{x}} = \sum_{n \in \mathcal{A}} w_n \mathbf{b}_n \tag{6}$$

In the case that all the atoms contain exact zeros, a small offset ϵ is added to $\mathbf{x}$ in order to avoid divisions by zero or taking the logarithm of zero in the further processing steps.

A. Initialization

Before any further processing, each dictionary atom is normalized to Euclidean unit length. The normalization was found to speed up the convergence of the algorithm, since otherwise the addition of new atoms to the active set would be dependent on their scale.

After the normalization, the set of active atoms is initialized with a single index n that alone minimizes the KL divergence as

$$\mathcal{A} = \{\operatorname*{argmin}_n \mathrm{KL}(\mathbf{x}||w_n\mathbf{b}_n)\}, \tag{7}$$

where the weight of each atom that minimizes the divergence separately is given as [17]

$$w_n = \frac{\mathbf{1}^T\mathbf{x}}{\mathbf{1}^T\mathbf{b}_n}. \tag{8}$$

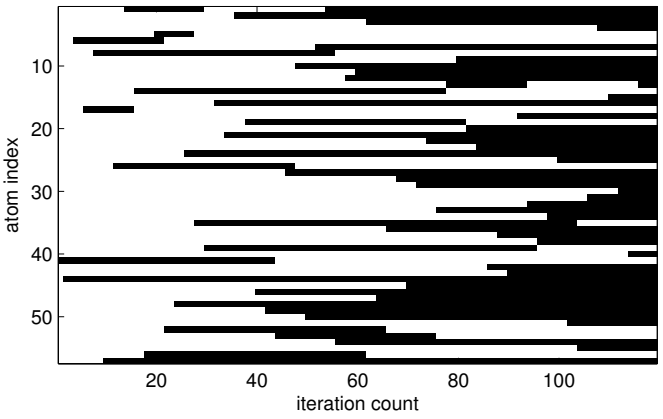

Fig. 1. The set of active atoms per each iteration for an example observation vector that is the magnitude spectrum of a mixture of two speakers. Black color indicates active atoms, and white indicates non-active atoms. The dictionary consists of 10 000 exemplars from individual speakers, but only the weights of those atoms are illustrated which were active at some point in the execution of the algorithm. New bases are added every second iteration, i.e., $K = 2$.

Here $\mathbf{1}$ is a all-one vector of length F.

We found that convergence of the proposed method was faster with the above initialization than using the gradient-based method that is used to add the other atoms.

B. Adding bases to the active set

Every K-th iteration starting from the first one, one atom is added to the set of active bases. As will be explained in the next section, the weights are estimated using the Newton algorithm, which is guaranteed to converge, if the function to be minimized is locally close to quadratic. It was found out that adding active atoms every iteration made the algorithm oscillate, and therefore having $K > 1$ is required.

The atom whose weight derivative is the lowest among the atoms not already in the active set, i.e., the atom which will decrease the KL divergence the most, is added. If the smallest derivative is negative, the index set is updated as

$$\mathcal{A} \leftarrow \mathcal{A} \cup \left\{\operatorname*{argmin}_{n \notin \mathcal{A}} \frac{\delta}{\delta w_n}\mathrm{KL}(\mathbf{w})\right\}, \tag{9}$$

with $\mathrm{KL}(\mathbf{w})$ defined as the KL-divergence as a function of the weight vector

$$\mathrm{KL}(\mathbf{w}) \equiv \mathrm{KL}(\mathbf{x}||\mathbf{B}\mathbf{w}) \tag{10}$$

and its derivative with respect to w_n is given as

$$\frac{d}{dw_n}\mathrm{KL}(\mathbf{w}) = \mathbf{b}_n^T(\mathbf{1} - \frac{\mathbf{x}}{\hat{\mathbf{x}}}) \tag{11}$$

Here the division of vectors is calculated entry wise, and $\hat{\mathbf{x}}$ is computed according to (6). The weight of the added atom is initialized to a small positive value ϵ_0. If the lowest derivative in (9) is positive, no new atoms are added.

As explained in the next section, the proposed method also allows *removing* atoms from the active set. Figure 1 illustrates the active atoms as the function of the iteration count for the data that is used in the evaluations in Section VI. Only two of the ten atoms that were selected in the first 20 iterations appear in the final, optimal set of active atoms.

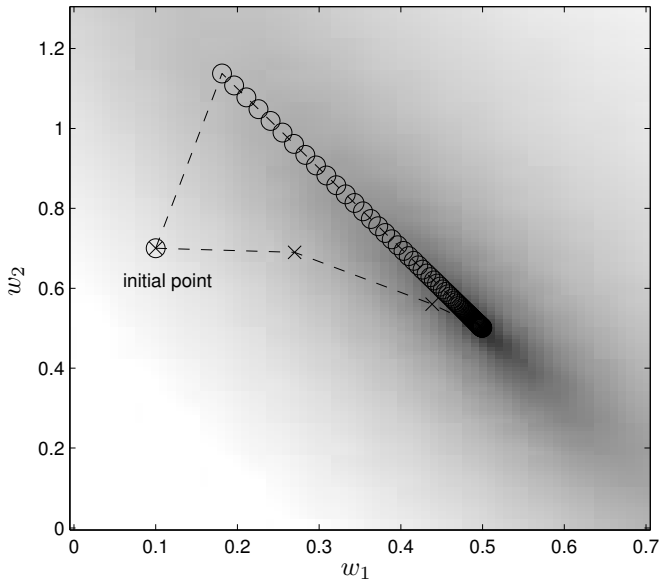

Fig. 2. The value of the KL divergence (higher intensity meaning a smaller value) as the function of two weights w_1 and w_2, with the two dictionary atoms $\mathbf{b}_1 = [1\ 1]^T$ and $\mathbf{b}_2 = [1\ 3]^T$, and the observation vector $\mathbf{x} = [1\ 2]^T$. The diagonal structure in the figure shows that w.r.t the KL divergence, w_1 and w_2 are strongly negatively correlated. The figure also shows estimates of w_1 and w_2 given by the EM algorithm (circles) and the Newton method (crosses) for multiple iterations, with both algorithms are initialized at the point $w_1 = 0.1$, $w_2 = 0.7$.

C. Updating weights

The weights of the atoms in the active set $\mathcal{A}$ are updated by the Newton method, where the step size is chosen so that the resulting weights are non-negative. The motivation for the use of the Newton method stems from the observation that in the overcomplete case, the dictionary atoms are inevitably correlated with each other. Therefore, the values of the KL divergence (3) as the function of weights (10) of different atoms are also correlated. Even in a non-overcomplete case the KL divergence as the function of the weights is correlated, as shown by a simple example in Fig. 2.

Existing methods for estimating the weights based on first-order optimization do not take into account linear dependencies of variables, and therefore require several iterations to reach the global optimum. On the other hand, the Newton algorithm is able to reach the optimum fast, provided that the surface of the function is approximately quadratic near the optimum. The convergence of a first-order optimization method (the EM algorithm) and the Newton method is also illustrated in Fig. 2. Unlike our method, the existing second-order methods [19]–[21] calculate Hessian matrices for all the atoms, which becomes infeasible when overcomplete dictionaries are used, or calculate the second derivative of only a single weight [22], [23], which does not allow modeling cross-correlations between atoms.

The update of the weights proceeds as follows. Let us denote a dictionary matrix whose columns consists of atoms in the active set $\mathcal{A}$ as $\mathbf{B}_\mathcal{A}$, and a weight vector which consists of weights of the active atoms as $\mathbf{w}_\mathcal{A}$, so that the model (6) can be written as $\hat{\mathbf{x}} = \mathbf{B}_\mathcal{A}\mathbf{w}_\mathcal{A}$. The gradient of the KL divergence (3) with respect to the weight vector $\mathbf{w}_\mathcal{A}$ of active atoms, computed at $\mathbf{w}_\mathcal{A}$ is given as

$$\nabla_{\mathbf{w}_\mathcal{A}} = \mathbf{B}_\mathcal{A}^T(\mathbf{1} - \frac{\mathbf{x}}{\hat{\mathbf{x}}}), \tag{12}$$

and the Hessian matrix with respect to $\mathbf{w}_\mathcal{A}$, computed at $\mathbf{w}_\mathcal{A}$ is given by

$$\mathbf{H}_{\mathbf{w}_\mathcal{A}} = \mathbf{B}_\mathcal{A}^T \operatorname{diag}(\frac{\mathbf{x}}{\hat{\mathbf{x}}^2})\, \mathbf{B}_\mathcal{A}. \tag{13}$$

Here, "diag" denotes a diagonal matrix whose diagonal entries consists of its argument vector, and $\hat{\mathbf{x}}^2$ denotes entry-wise squaring of vector $\hat{\mathbf{x}}$. The weights are updated as

$$\mathbf{w}_\mathcal{A} \leftarrow \mathbf{w}_\mathcal{A} - \alpha\mathbf{p}, \tag{14}$$

where α is the step size, and $\mathbf{p}$ is the search direction given as

$$\mathbf{p} = (\mathbf{H}_{\mathbf{w}_\mathcal{A}} + \epsilon\mathbf{I})^{-1}\nabla_{\mathbf{w}_\mathcal{A}} \tag{15}$$

An identity matrix $\mathbf{I}$ multiplied by a small constant $\epsilon = 10^{-10}$ is added to the Hessian matrix before calculating its inverse to ensure the numerical stability of the inversion.

We calculate the ratio vector $\mathbf{r} = \mathbf{w}_\mathcal{A}/\mathbf{p}$ by element-wise division, and obtain the step size parameter α as

$$\alpha = \min_{r_i>0}\ r_i. \tag{16}$$

If α is larger than 1, step size $\alpha = 1$ is used, which corresponds to the standard Newton algorithm. Estimating the step size as in (16) ensures that the weights resulting from (14) are non-negative. If a weight becomes zero as the result of (14), the corresponding index will be removed from the set of active bases.

D. Termination

As we will shown in Section IV-A, $\mathrm{KL}(\mathbf{w})$ is convex, and therefore the global optimum satisfies [42, p. 142]

$$\left[\frac{\delta}{\delta w_i}\mathrm{KL}(\mathbf{w})\right] w_i = 0 \tag{17}$$

$$\frac{\delta}{\delta w_i}\mathrm{KL}(\mathbf{w}) \geq 0 \tag{18}$$

$$w_i \geq 0. \tag{19}$$

The algorithm is terminated when all the derivatives in (11) are larger than predefined threshold ϵ_δ, and when the norm of the gradient in (12) is smaller than threshold ϵ_∇. As a result, the derivative of the weights in the active set is zero, and the derivatives of weights not in the active set are positive. Together these conditions make the algorithm fulfill the optimality conditions in Eqs. (17) - (19).

E. Summary of the proposed algorithm

The proposed estimation algorithm is iterative, and is summarized as follows.

1: **Initialize**
2: $iteration\ count = 0$

Normalize each dictionary atom to unity norm.
Calculate all the weights using (8).
Select the first active atom using (7).
repeat
Update $\hat{\mathbf{x}}$ according to (6)
if $iteration\ count$ mod K = 0 **then**
Calculate the gradient of the KL w.r.t weights that are not in the active set using (11).
If the weight with the smallest derivative is negative, add the corresponding atom to the active set and set its weight to ϵ_0.
end if
Calculate the gradient of the KL divergence w.r.t the weights of the active set using (12).
Calculate the Hessian matrix using (13), the search direction using (15), and the step size using (16).
Update the weights using (14). If a weight becomes zero, remove the corresponding atom from the active set.
Set $iteration\ count = iteration\ count + 1$
until the stopping criteria given in Section III-D are fulfilled.

IV. Analysis

In this section we show that the weight vector of a non-negative overcomplete representation is sparse, and that there is a unique representation $\hat{\mathbf{x}}$ which minimizes the KL divergence between an observation and the model. Additionally, we analyze the computational complexity of the proposed algorithm and its ability to process multiple observations. The conclusions about the sparsity and uniqueness of the non-negative overcomplete representation also apply to other divergences that are convex, and the principle of the proposed algorithm can be applied with other convex divergences.

A. Convexity and Uniqueness

Let us first consider the KL divergence as a function of $\hat{\mathbf{x}} = \mathbf{B}\mathbf{w}$. It is rather trivial to show that $\mathrm{KL}(\mathbf{x}||\hat{\mathbf{x}})$ is strictly convex in $\hat{\mathbf{x}}$, and that the feasible set $\{\hat{\mathbf{x}} = \mathbf{B}\mathbf{w}|\mathbf{w} \geq \mathbf{0}\}$ is convex. We can now state the following lemma.

Lemma 1. *The KL divergence* $\mathrm{KL}(\mathbf{w}) \equiv \mathrm{KL}(\mathbf{x}||\mathbf{B}\mathbf{w})$ *is convex in* $\mathbf{w}$. *Moreover, if the columns of* $\mathbf{B}$ *are linearly independent,* $\mathrm{KL}(\mathbf{w})$ *is* strictly *convex in* $\mathbf{w}$.

The proof of Lemma 1 is presented in Appendix A. As a consequence of the Lemma, $\mathrm{KL}(\mathbf{x}||\hat{\mathbf{x}})$ has a unique global minimum in $\hat{\mathbf{x}}$. Also, for $\hat{\mathbf{x}} = \mathbf{B}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}}$, the divergence, as a function of $\mathbf{w}_{\mathcal{A}}$, has a unique minimum value for any $\mathbf{B}_{\mathcal{A}}$. In addition, if the columns of $\mathbf{B}_{\mathcal{A}}$ are linearly independent, the value of $\mathbf{w}_{\mathcal{A}}$ that reaches this minimum is also unique.

B. Sparsity

In the proposed algorithm we do not constrain the number of active atoms in any way. However, it turns out that non-negative overcomplete representations are sparse even without additional constraints. This is important, as sparsity enables applications such as sparse coding [8], [9] and has been shown beneficial in applications such as source separation [3].

Lemma 2. *Let* $\mathbf{w}^*$ *be a weight vector that reaches the global minimum of KL with* L *non-zero entries,* i.e. $||\mathbf{w}^*||_0 = L$. *Let* $\mathbf{B}_{\mathcal{A}}$ *be a matrix whose columns are the atoms which correspond to the* L *non-zero entries of* $\mathbf{w}^*$, *and let* $\mathbf{w}^*_{\mathcal{A}}$ *be the corresponding weight vector which consists of non-zero entries of* $\mathbf{w}^*$. *If the columns of* $\mathbf{B}_{\mathcal{A}}$ *are linearly dependent, there always exists a non-negative weight vector* $\mathbf{w}$ *such that* $||\mathbf{w}||_0 < ||\mathbf{w}^*_{\mathcal{A}}||_0$ *that also achieves the globally optimal error over* $\mathbf{B}_{\mathcal{A}}$.

The proof of the lemma is straightforward and given in Appendix B.

Lemma 3. *If the algorithm is allowed to converge for each* $\mathcal{A}$, *the active dictionary atoms in* $\mathcal{A}$ *are always linearly independent.*

The proof of the lemma is given in Appendix C. In practice we only perform K iterations of the Newton update for any $\mathcal{A}$, rather than letting it run to convergence. Nevertheless, the set of active atoms generally remains linearly independent.

As a consequence of the above lemmas, the active set selected by the algorithm will naturally be non-redundant and therefore sparse even though no active sparsity constraint is employed (although it may not necessarily arrive at the optimally sparse active set for a given error). The maximum number of linearly independent atoms in a dictionary is the dimensionality F. As a result, it is possible to minimize the KL divergence with a weight vector having at most F non-zero weights. The evaluations in Section VI show that in practical situations the number of non-zero entries is much fewer than F, demonstrating that this upper bound for the number of active atoms is rather loose. Explicit sparseness constraints [43] could be used to find sparser representations that have either the same or maybe slightly higher divergence.

C. Rate of convergence of algorithms

Active set selection algorithms such as the proposed algorithm can be shown to have linear convergence rates under explicit conditions of sparsity when the objective functions have bounded Hessians, *i.e.* they satisfy quadratic-like conditions of smoothness and convexity [28], [29]. Unfortunately, the KL divergence does not fall in this category of objective functions, since its convexity approaches that of first-order linear functions on the one end and is exponential on the other, and its second derivatives cannot be bounded. In practice, however, for any finite dictionary, and for bounded inputs, the diagonal entries of the Hessian are bounded from above. Moreover, we add a diagonal term to the Hessian, effectively applying an additional ℓ_2 regularization term to the objective function. Consequently, the convergence rate can be expected to be linear as well, although rigorous demonstration of this is outside the scope of this paper. As will be shown later in the paper, experimental results confirm this expectation.

D. Asymptotic complexity of algorithms

The convergence rate of the proposed algorithm is analyzed using computational simulations in Sections V and VI. In this section we briefly study the asymptotic complexity of the proposed method and compare it to the EM algorithm as a function the number of features F and the size of dictionary N. Let us first study the computational complexity of each iteration, and denote the number of atoms in the active set by A. Computationally the most complex operations of the proposed algorithm are:

- updating the model in (6): complexity $\mathcal{O}(FA)$
- calculation of the derivative in (11): complexity $\mathcal{O}(FN)$
- calculation of the Hessian matrix in (13): $\mathcal{O}(AF^2)$
- calculation of the inverse of the Hessian matrix in (15): complexity $\mathcal{O}(A^3)$

In the overcomplete case we will have $F < N$. We have also shown that $A \leq F$, so that the sum of the above operations is $\mathcal{O}(FN + A^3)$, which is the complexity of the algorithm per iteration.

In an experimental evaluation the calculation of the gradient was found to be the most time-consuming operation. Its asymptotic complexity is fixed per iteration. For a fixed number of iterations I, we can therefore approximate the asymptotic complexity of the proposed method as $\mathcal{O}(IFN)$.

The reference EM algorithm is composed of matrix-vector products and element-wise products and divisions of vectors, and it is easy to see that its most time-consuming operation is $\mathcal{O}(FN)$. Assuming a fixed number of iterations I, its total complexity is therefore also $\mathcal{O}(IFN)$. Since the evaluations in Section VI show that the proposed method requires much fewer iterations to converge than the EM algorithm (which, in fact, does not seem to converge in a finite number of iterations), the proposed method is much faster. Values of the above parameters found in previous studies have the following ranges: $F = 5...1\,000$, $N = 3...100\,000$, and $I = 50...600$ (for the EM algorithm).

E. Processing multiple observations

The algorithm description in Section III was derived for a single observation vector. In practical usage situations, we are often interested in deriving the representation for multiple observation vectors. For example in audio signal processing applications, an observation vector is calculated from short-time frames of the signal, and the representation needs to be calculated for each frame.

The representation is independent for each observation, meaning that the parameters of one observation are not affected by other observations. If the same dictionary is used for multiple observations, calculating the representation for all of them at once is more efficient, since some of the operations that are matrix-vector products for a single observation can then be expressed as a matrix-matrix products. These operations include at least the calculation of the model (6) and the gradient (12). For example, in the MATLAB implementation that we used, the use of matrix-matrix products was between three to ten times faster than repeated matrix-vector products, depending on the sizes of the matrices. Also the EM algorithm benefits significantly from parallel processing, since all its operations can be expressed as matrix products or divisions.

It should be noted that not all the operations of the proposed method can be easily parallelized. The set of active basis is separate for each observation, and therefore the Hessian matrix is different for each frame.

V. Evaluation setup

The proposed method was compared to conventional NMF algorithms in representing mixtures of speech magnitude spectrograms.

A. Acoustic data

As the acoustic data we use the subset of the GRID corpus [44] that was used as the training set in the Speech Separation Challenge [45]. The corpus consists of speech from 34 different speakers. There are 500 sentences from each speaker, and each sentence consists of simple sequences of six words. The sampling frequency of the signals is 25 kHz.

For evaluation, we generated a test set of 100 signals, each of which is a mixture of signals from two speakers. Each test signal is generated by picking two random speakers and a random sentence from both speakers. The shorter of the two signals is zero-padded to make their lengths equal. The signals are scaled to equal root-mean-square levels and summed.

All the data is represented as the short-time magnitude spectra of the signals, which is the standard representation for audio content analysis and spectrogram factorization based source separation. As in [5], we window the signals using a 60 ms Hanning window with a 15 ms window shift. The magnitude spectrum of each frame is calculated by taking the discrete Fourier transform (DFT) and calculating the absolute value of each entry. The resulting non-negative DFT feature vector is of length $F = 751$ for each frame.

B. Dictionaries

In addition to evaluating the convergence of the algorithm, we evaluated the capability of the representation to separate sources. Therefore we construct dictionaries so that there is a separate set of atoms for each speaker. The atoms were generated for each speaker by clustering the training data of the speaker, which consists of all the sentences from the speaker, excluding those sentences that were used to generate the mixture signals.

For dictionary generation, the magnitude spectra were normalized to sum to unity, since want the dictionaries to be gain-independent. The spectra were clustered into N clusters by k-means clustering. In order to make the clustering match the divergence used in the representation, each spectrum was assigned to the cluster center that minimizes the divergence (3) between the spectrum and the cluster center. New cluster centers were obtained iteratively as the mean of the spectra assigned to each cluster. A random subset of 30 000 frames per speaker of the training data was used.

Dictionaries have been previously obtained by clustering e.g. in [36], [37]. We tested also randomly sampled dictionaries [3], but their source separation performance was

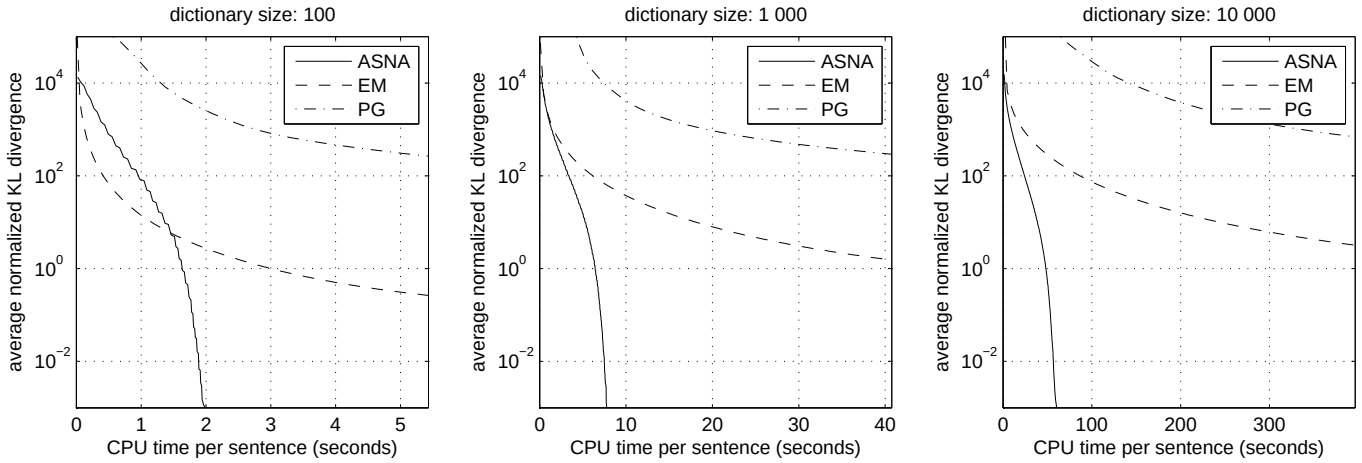

Fig. 3. The average normalized KL divergence per sentence as the function of the cumulative CPU time.

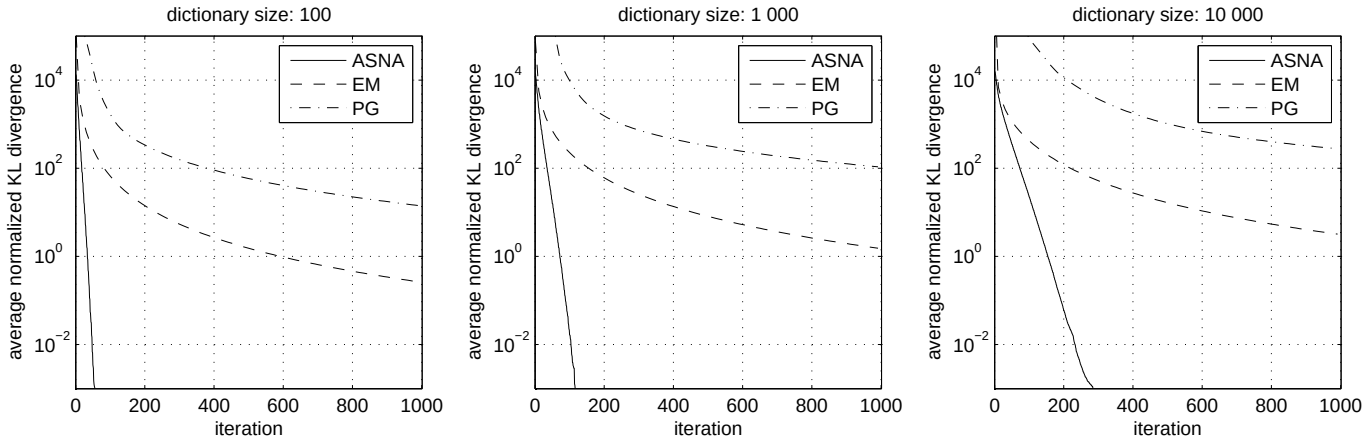

Fig. 4. The average normalized KL divergence per sentence as the function of the iteration count.

slightly lower and therefore the evaluation is done with the clustered dictionaries. Parts-based dictionaries [46] that can be obtained e.g. with the NMF were not used, since learning overcomplete parts-based dictionaries would require careful tuning of additional constraints such as sparsity.

Three different dictionary sizes were evaluated: 50, 500, and 5 000 atoms per speaker. For each of the dictionary sizes, the clustering of the spectra was done separately. For representing a test signal, we combine the dictionaries of the speakers in the test signal to form a single dictionary. This results in dictionary sizes of 100, 1 000, and 10 000 atoms. Note that the smallest dictionary is only included for comparison as it is not overcomplete.

C. Evaluated methods

The proposed ASNA method is compared to two methods that are available for minimizing the KL divergence with the non-negativity constraints: EM algorithm and projected gradient [11, pp. 267-268]. The EM algorithm was briefly explained in Section II-C. It is initialized with an all-one weight vector, and consists of repeated use of update equation (5). As with ASNA, the atoms were normalized to unity norm for the EM algorithm, which was found to speed up its convergence.

The projected gradient (PG) method consists of updating the weight vector towards its negative gradient, and setting negative weights to zero after each update. The step size was updated with a variant of the bold driver algorithm [47], so that the step size was halved until the divergence was decreased, and after each decrease of the divergence the step size was increased by 5%. The step size was updated separately for each frame of the test sentences. The weights of the PG method were initialized to $w_i = (\mathbf{x}^T\mathbf{b}_i)/(\mathbf{1}^T\mathbf{b}_i)$, which was found to produce faster convergence in comparison to e.g. using the single-atom initialization of the proposed method.

The evaluation was run on a desktop computer with an Intel 2-core Duo E8500 3.16GHz processor. Both the tested methods were implemented using MATLAB version R2010b, with multi-core processing enabled. A double-precision accuracy was used to represent all the variables. The set of active atoms in the proposed method was implemented using the sparse matrix datatype, which does not allocates memory for the whole weight matrix, but only for its non-zero weights and their indices. A reference implementation of the proposed method is available at http://www.cs.tut.fi/~tuomasv.

For ASNA we used $K = 2$, meaning that new entries were added to the set of active atoms every second iteration. Different values of K were briefly tested on a development data that was not part of the test material. With $K = 1$ the proposed method did not converge, but started oscillating. The reason for this is that after adding a new atom to the active set, a single weight update is not enough to make the function to be minimized close enough to quadratic for the next iterations. With $K = 3$ the proposed method converged, but slower than with $K = 2$ because new atoms were added to the active set less often. The initial weights of added atoms were set to $\epsilon_0 = 10^{-15}$ The stopping criteria for ASNA, i.e., the threshold of the weight derive for adding new atoms was set to $\epsilon_\delta = -10^{-15}$ and the norm of the gradient was set to $\epsilon_\nabla = 10^{-15}$. The above values were chosen based on experiments with observations and dictionaries not used in the final experiments. The proposed method was executed until it converged (according to the criteria in Section III-D). For the EM and PG algorithms, the maximum number of iterations was set to 1 000.

VI. Evaluation metrics and results

We evaluate three different aspects of the algorithms: their ability to minimize the KL divergence and the used CPU time, the sparseness of the resulting representation, and the source separation capability of the representation derived with the algorithms. The metrics and results are given in the following subsections.

A. Convergence and CPU time usage

As an evaluation metric for the convergence of the algorithms we use the cumulative KL divergence over all the frames of each signal, averaged over all the signals in the test set. All the tested algorithms are deterministic and were therefore executed only once for each combination of a test signal and a dictionary size. The total number of frames (i.e., observation vectors) in all the test signals was 12 973. Since there were multiple dictionaries involved (a separate dictionary for each speaker) and the divergence is calculated over a large number of frames, only a single random draw of the dictionaries was used instead of repeating the experiment multiple times.

The minimum KL divergence achievable is different for each dictionary. In order to make the results with different dictionary sizes comparable with each other, we normalize the results by subtracting the globally minimal KL divergence for each particular dictionary that was used — which is the KL divergence for the proposed method after it has converged. This is dubbed the "normalized KL divergence".

The average normalized KL divergence as the function of cumulative CPU time for different dictionary sizes is illustrated in Fig. 3. The projected gradient method is in all the cases worse than the other two methods, so we focus on comparing the proposed ASNA method with the EM algorithm. When a large enough CPU time is used, ASNA is able to reach significantly lower values of the normalized KL divergence in comparison to the EM algorithm for all

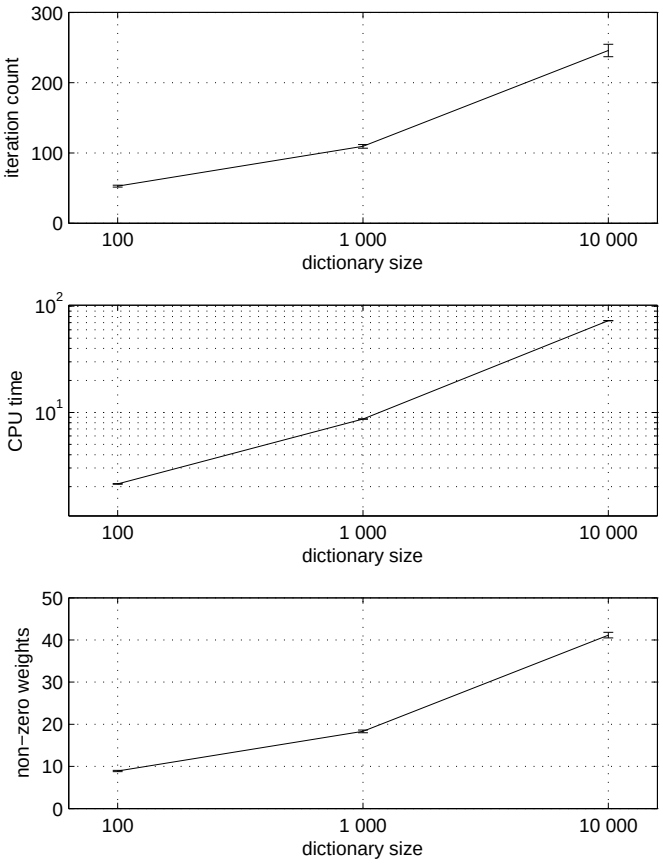

Fig. 5. The average number iterations required for the proposed algorithm to converge (upper panel), the average CPU time per sentence required to converge (middle panel), and the average number of non-zero weights in the optimal solution (bottom panel) as the function of the dictionary size. The vertical bars denote 95% confidence intervals of the averages.

dictionary sizes. For dictionary sizes 1 000 and 10 000, ASNA always achieves lower divergences for the same cumulative CPU times. Only for dictionary size 100 and a CPU time less than 1.5 seconds, the EM algorithm performs better than ASNA. The 100-atom dictionary is non-overcomplete and therefore not a target of this paper, but in this case the good relative performance of the EM algorithm in comparison to ASNA can be attributed to the ratio of active atoms and the dictionary size being higher, which is higher for smaller dictionaries (as will be shown later).

Even though the convergence of the EM algorithm is fast in the beginning, for all the dictionary sizes its asymptotic performance is much slower than that of the proposed method. In the case of the 10 000-atom dictionary, ASNA is able to reach the lowest value of the normalized KL divergence obtained with the EM algorithm approximately eight times faster than the EM algorithm. All the tested methods decrease the value of the normalized KL divergence monotonically. It can be observed that the ASNA decreases the KL-divergence step-wise (visible at least for dictionary size 100). This is caused by the addition of new atoms to the active set every second iteration, after which the first Newton update decreases the divergence more than those iterations where new atoms are not added to the active set.

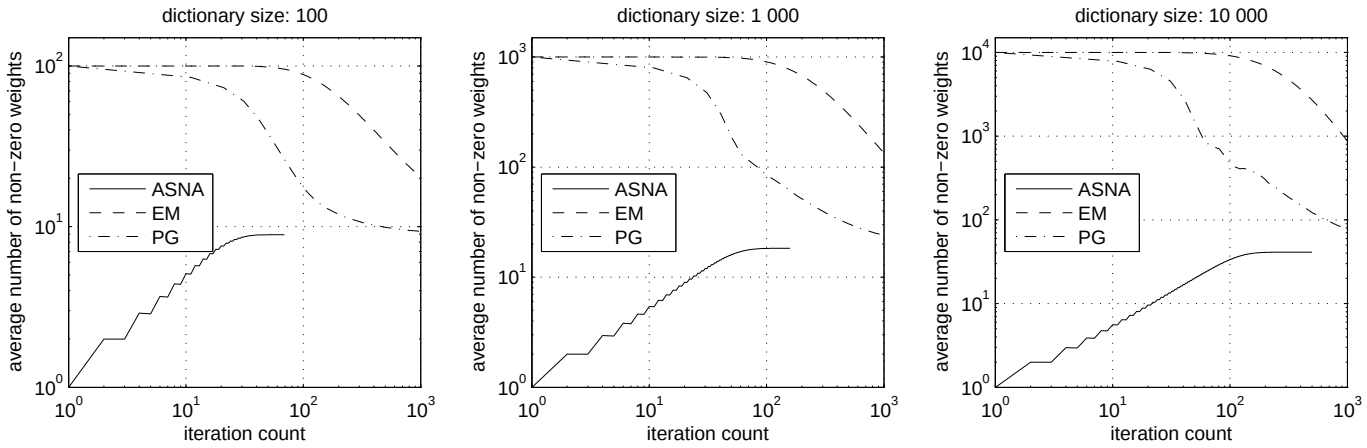

Fig. 6. The average number of non-zero atoms per each frame as the function of the iteration count.

Figure 4 shows the average normalized KL divergence as the function of the iteration count. We can observe that the rate of convergence of ASNA is approximately linear. In general, Newton methods converge quadratically, if the function to be minimized is approximately quadratic near the global minimum. In the proposed method, however, the convergence is not quadratic since new atoms are added to the active set every second iteration. Once the dictionary is fixed, the algorithm finds the global optimum in a few iterations. This final convergence behavior is not properly visible in Figure 4, since the iteration where the optimum is reached is different for each test frame.

The two top panels of Fig. 5 illustrate the average number of iterations required for ASNA to converge and the average CPU time required to converge. Even though the number of different combinations of active atom vectors increases exponentially with the dictionary size, the upper panel of the figure shows that ASNA is able to find the optimal set of active bases in a rather small number of iterations. Moreover, the number of iterations increases only sublinearly as the function of the dictionary size. While the number of samples may be too limited to draw definite conclusions, these results indicate that the number of iterations required to converge is approximately linearly dependent on the logarithm of the dictionary size.

The relationship between the CPU time and the dictionary size is approximately linear on a logarithmic scale. The slope of the line, however, is smaller than one, which shows that the computation time grows sublinearly as the function of the dictionary size. Based on the analysis in Section IV-D the calculation of the weight derivative is linearly dependent on the dictionary size. Therefore the sublinearity originates from the number of iterations, which grows sublinearly as the function of the dictionary size. The average amount of CPU time required for the ASNA to converge was 16, 67, and 563 milliseconds per frame for dictionary sizes 100, 1 000, and 10 000, respectively.

B. Sparseness

The bottom panel of Fig. 5 illustrates the average number of non-zero weights in the optimal solution as the function of the dictionary size. Since the dictionary atoms are linearly independent ($\mathbf{B}$ is full rank) and the dimensionality of the atoms $F = 751$, the theoretical limits for the number of non-zero weights derived in Section IV-B in the case of 100, 1 000, and 10 000 atom dictionaries are 100, 751, and 751, respectively. We can observe that the average number of non-zero weights for each dictionary size is approximately linear as a function of the logarithm of the dictionary size, and that the obtained number of non-zero weights is much smaller than the derived theoretical limits: 9, 18 and 41 atoms for the 100, 1 000, and 10 000 atom dictionaries, respectively.

Figure 6 illustrated the average number of non-zero weights as the function of the iteration count for the tested methods. In the case of the EM algorithm the weights do not necessarily become exactly zero, and therefore values below 10^{-15} were considered to be zero. All the methods converge towards the same value. For ASNA this happens by increasing the number of active atoms, whereas for EM and PG it happens by decreasing the number of atoms. The figure shows that EM is very inefficient in producing exact zeros, even when a large number of iterations is used.

In order to test how the complexity of the acoustic material affects the sparseness of the representation, we created also random mixtures of one to ten speakers. The mixture signals and the dictionaries were generated in the same manner as the mixtures described in Section V for two speakers. 100 mixtures per each number of speakers were used. Two different types of dictionaries were tested: 1) dictionaries with 500 atoms per speaker, and 2) dictionaries where the total number of atoms was 500, i.e., the atoms per each speaker was 500 divided by the number of speakers.

Figure 7 illustrates the average number of active atoms as the function of the number of speakers for the two different dictionary types. In the case of the 500-atom dictionary, the number of active atoms increases slightly as the number of speakers increases. However, increases is less than 50% when switching from one-speaker signals to ten-speaker mixtures. When 500 atoms per speaker are used, the number of atoms increases sublinearly as the function of the number of speakers. However, most of this increase can be attributed to the increase

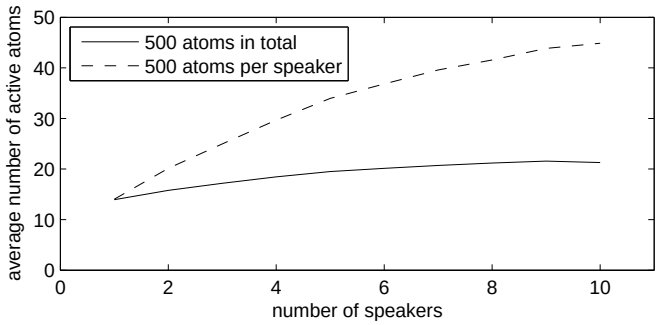

Fig. 7. The average number of active atoms as the function of the number of speakers.

of the dictionary size, and actually the relative amount of active atoms increases as the number of speakers increases.

C. Source separation performance

We are also interested how the non-negative representations obtained by the proposed algorithm affect the performance on real applications. Source separation [33], [48], the process of estimating the individual source signals that make up a mixture, is one of the most commonly used applications of non-negative representations. The majority of the existing source separation algorithms based on non-negative representations use the EM algorithm for estimating the model parameters [2]–[7], [15], [17], [18], [34], [38], and only the way the dictionaries are constructed or the atoms are represented differ. ASNA gives solutions identical to the EM algorithm (provided that they are run for enough iterations), and can therefore replace the EM algorithm in source separation systems that use a fixed dictionary, to give a faster convergence.

The source separation performance was evaluated using the two-speaker mixtures described above. Since the dictionaries used to represent the mixture signal consists of two sets of speaker-specific atoms, we can use the representations derived in the previous section to do source separation. By denoting the sets of atoms for speaker 1 and 2 by $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively, we design frequency-domain filter vectors $\mathbf{h}_i$ for source $i \in 1, 2$ as

$$\mathbf{h}_i = \frac{\sum_{n \in \mathcal{A}_i} w_n \mathbf{b}_n}{\sum_{n \in (\mathcal{A}_1 \cup \mathcal{A}_2)} w_n \mathbf{b}_n}$$

The complex spectrogram of source i is then obtained by point-wise multiplying the magnitude DFT spectrum of the mixture signal and the above filter, and copying the phases from the mixture spectrum. A time-domain signal in the frame is obtained by inverse discrete Fourier transform, and the frames are combined by the weighted overlap add [49, Chapter: Overlap-Add STFT Processing].

We evaluate the quality of the separated signals by calculating the signal-to-distortion ratio (SDR) between the separated signal $\hat{s}(t)$ and the original source signal $s(t)$ before mixing. The SDR in dB is obtained as

$$\mathrm{SDR}_{\mathrm{dB}} = 10 \log 10 \frac{\sum_t s(t)^2}{\sum_t (s(t) - \hat{s}(t))^2}.$$

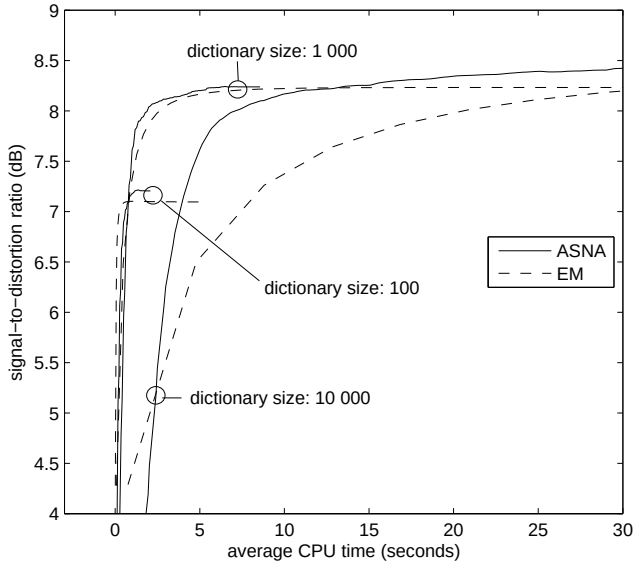

Fig. 8. The average signal-to-distortion ratio for separated sources as the function of cumulative CPU time per sentence. The results are shown separately for each dictionary size.E

The SDRs over both the mixed speakers and all the test cases are averaged.

The average SDRs for the proposed ASNA method and the EM algorithm for different dictionary sizes as the function of the used CPU time are illustrated in Fig. 8. PG reached high SDRs slower than the other two methods, and is not included in the figure in order to retain its clarity. Since the SDRs of the mixtures are 0 dB, it is clear that both the tested algorithms are able to significantly improve the SDR, and that increasing the dictionary size increases the source separation performance. For a specific dictionary size, both the methods saturate to equal SDRs. While for the smaller dictionaries, ASNA does not perform significantly better than the EM algorithm, for the largest dictionary (10 000 atoms) ASNA reaches the highest SDR much faster than the EM algorithm. Since ASNA requires a few iterations to add a sufficient number of atoms to the active set, it gives poor performance if less than 2 seconds of processing time is used with the largest dictionary.

Even though ASNA converges to lower KL-divergences, this does not seem to translate to higher SDRs. This is most likely because the separation problem here is rather simple; there are only two sources, and finding the exact optimum is not needed. However, being able to reach lower KL divergences has been found beneficial in other applications of non-negative overcomplete representations where the problem to be solved involves estimating more than two factors. For example, when the weights are used to estimate likelihoods for a large number of states, achieving lower KL-divergences by increasing the number of EM algorithm iterations from 200 to 600 was found to give a significant improvement [50].

VII. Conclusions

This paper has proposed an efficient algorithm, ASNA, for estimating the weights of a non-negative overcomplete repre-

sentation, where an observation is modeled as the weighted sum of atoms in a dictionary. The criterion for estimating the weights is the minimization of the Kullback-Leibler divergence, which is more appropriate for audio than e.g. the Euclidean distance. ASNA is based on updating an active set of atoms iteratively, and estimating the weights of atoms in the active set by the Newton method, where the step size is estimated to retain the non-negativity of the weights.

The performance of ASNA was compared to the state-of-the-art EM algorithm and projected gradient. We show that in representing magnitude spectra of audio signals, ASNA is able to reach lower values of the KL divergence in comparison to the reference methods. On large dictionaries the atom weights produced by ASNA always lead to lower KL divergences than the other methods. Moreover, on large dictionaries, ASNA is able to reach the lowest KL divergence values attainable by the EM algorithm approximately eight times faster. We show that the non-negative overcomplete representations are sparse, meaning that only a minority of the weights are non-zero. We also show that ASNA is able to find the optimal set of non-zeros atoms in a relatively small number of iterations.

The evaluation on audio source separation shows that larger dictionaries lead to better separation, and ASNA reaches the highest attainable separation quality with the least computation time.

Regarding future work it should be noted that the most time-consuming part of the proposed algorithm is the calculation of the derivative of the KL divergence with respect to all the weights, which is required to find a new atom to add to the active set. Future work therefore includes the development of methods which calculate these derivatives only for a subset of the atoms at each iteration, for example by first clustering the dictionary as in [37].

The proposed method uses the basic Newton method, which explicitly calculates the inverse of the Hessian matrix. Another way to speed up the algorithm would be to use quasi-Newton methods, which avoid the calculation of the Hessian or its inverse.

The tested method uses a fixed value $K = 2$. The first iterations where the cost function is locally less quadratic are likely to require two Newton steps to update the weight, but it may be possible to develop strategies where K is adapted depending on the shape of the cost function, or even where multiple new atoms are added at once.

Finally, even though we showed that the non-negativity constraints alone lead to sparse representations, including explicit sparseness constraints may provide advantages in some applications. If the constraints are differentiable, it will be possible to include them by simply changing the gradients and Hessian matrices used in ASNA.

APPENDIX A
CONVEXITY OF THE KL DIVERGENCE

We will employ the following definition of convexity: a function $f(\mathbf{x})$ is convex in $\mathbf{x}$ if $f(\mathbf{x}) - f(\mathbf{x}_1) \geq \nabla_{\mathbf{x}} f(\mathbf{x})^T (\mathbf{x} - \mathbf{x}_1)$ for any $\mathbf{x}$ and $\mathbf{x}_1$. If the left hand side of this inequality is strictly greater than the right hand side, the function is strictly convex and has a unique minimum in $\mathbf{x}$.

We have $\hat{\mathbf{x}} = \mathbf{B}\mathbf{w}$. We can write the KL divergence $\mathrm{KL}(\mathbf{w}) \equiv \mathrm{KL}(\mathbf{x}|\mathbf{B}\mathbf{w})$ as follows:

$$\mathrm{KL}(\mathbf{w}) = \mathbf{x}^T \log \frac{\mathbf{x}}{\mathbf{B}\mathbf{w}} - \mathbf{1}^T\mathbf{x} + \mathbf{1}^T\mathbf{B}\mathbf{w}$$

where the division and the logarithm of vectors is component wise. We also note that

$$\nabla_{\mathbf{w}} \mathrm{KL}(\mathbf{w}) = \mathbf{B}^T(\mathbf{1} - \frac{\mathbf{x}}{\mathbf{B}\mathbf{w}}) \tag{20}$$

Let us now consider $\mathrm{KL}(\mathbf{w}) - \mathrm{KL}(\mathbf{w}_1)$.

$$\begin{aligned} \mathrm{KL}(\mathbf{w}) - \mathrm{KL}(\mathbf{w}_1) &= \mathbf{x}^T \log \frac{\mathbf{B}\mathbf{w}_1}{\mathbf{B}\mathbf{w}} + \mathbf{1}^T\mathbf{B}(\mathbf{w} - \mathbf{w}_1) \\ &\geq \mathbf{x}^T(1 - \frac{\mathbf{B}\mathbf{w}}{\mathbf{B}\mathbf{w}_1}) + \mathbf{1}^T\mathbf{B}(\mathbf{w} - \mathbf{w}_1) \\ &= \nabla_{\mathbf{w}} \mathrm{KL}(\mathbf{w}_1)^T(\mathbf{w} - \mathbf{w}_1) \end{aligned} \tag{21}$$

giving us the desired relation for convexity.

In the above proof, we have invoked the relation $-\log(x) \geq 1 - x$, with equality occurring at $x = 1$. Correspondingly, the only condition under which the left hand side of Equation (21) equals the right hand side is when there exist $\mathbf{w}$ and $\mathbf{w}_1$ such that $\mathbf{B}\mathbf{w} = \mathbf{B}\mathbf{w}_1 \Rightarrow \mathbf{B}(\mathbf{w} - \mathbf{w}_1) = 0$, implying that the columns of $\mathbf{B}$ are linearly dependent. Conversely too, if the columns of $\mathbf{B}$ are linearly dependent, the two sides of the equation are equal for every pair of $\mathbf{w}$ and $\mathbf{w}_1$ whose difference lies in the null-space of $\mathbf{B}$.

Thus, the KL divergence is convex in $\mathbf{w}$, and, further, is *strictly* convex if the columns of $\mathbf{B}$ are linearly independent.

APPENDIX B
DENSITY OF OPTIMAL WEIGHT VECTOR OVER LINEARLY DEPENDENT SETS OF ATOMS

$\mathbf{B}_{\mathcal{A}} \in \mathcal{R}_+^{F \times L}$ is a matrix composed of L (F-dimensional) atoms, and $\mathbf{w}^*_{\mathcal{A}}$ is non-negative vector such that $||\mathbf{w}^*_{\mathcal{A}}||_0 = L$ which achieves minimum divergence in composing an observation vector $\mathbf{x}$ with $\mathbf{B}_{\mathcal{A}}$. In other words, if we define

$$\hat{\mathbf{x}}^* = \mathbf{B}_{\mathcal{A}}\mathbf{w}^*_{\mathcal{A}} \tag{22}$$

then

$$\hat{\mathbf{x}}^* = \min_{\hat{\mathbf{x}} = \mathbf{B}\mathbf{w} | \mathbf{w} \geq 0} \mathrm{KL}(\mathbf{x}||\hat{\mathbf{x}}).$$

If the columns of $\mathbf{B}_{\mathcal{A}}$ are linearly dependent, there exists a non-zero vector $\mathbf{z}$ so that $\mathbf{B}_{\mathcal{A}}\mathbf{z} = \mathbf{0}$. Let vector $\mathbf{r} = \mathbf{B}_{\mathcal{A}}/\mathbf{z}$, where the division is entry-wise. Without affecting our discussion, we will assume that division by 0 results in ∞. Let r_i be the i^{th} component of $\mathbf{r}$.

We can define a variable α as:

$$\alpha = \min_{r_i > 0} r_i$$

We can now construct a vector $\mathbf{w} = \mathbf{w}^*_{\mathcal{A}} - \alpha\mathbf{z}$ such that $\mathbf{w} \geq 0$, and $\hat{\mathbf{x}}^* = \mathbf{B}_{\mathcal{A}}\mathbf{w}$. By construction, $\mathbf{w}$ has at most $L - 1$ non-zero entries, *i.e.* $|\mathbf{w}|_0 \leq L - 1$. In other words, $\mathbf{w}$ is a non-negative vector that also results in the optimal estimate (since $\mathbf{B}_{\mathcal{A}}\mathbf{w} = \hat{\mathbf{x}}^*$) and has fewer non-zero entries than $\mathbf{w}^*_{\mathcal{A}}$. Also, by recursively applying the same construction, we can see that if the rank of $\mathbf{B}_{\mathcal{A}}$ is R, the sparsest optimal weight vector that we can obtain has no more than R non-zero entries, *i.e.* $|\mathbf{w}|_0 \leq R$.

APPENDIX C
LINEAR INDEPENDENCE OF EXPANDED ACTIVE SET

Let $\mathbf{B}_{\mathcal{A}}$ be a matrix composed from the current active set of atoms drawn from $\mathbf{B}$. Further, let all the columns of $\mathbf{B}_{\mathcal{A}}$ be linearly independent. Let $\mathbf{w}^*_{\mathcal{A}}$ be the unique optimal weight vector $\mathbf{w}_{\mathcal{A}}$ that minimizes $\mathrm{KL}(\mathbf{x}||\mathbf{B}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}})$, and let $\hat{\mathbf{x}}^*_{\mathcal{A}} = \mathbf{B}_{\mathcal{A}}\mathbf{w}^*_{\mathcal{A}}$. Since the columns of $\mathbf{B}_{\mathcal{A}}$ are linearly independent and the solution $\mathbf{w}^*_{\mathcal{A}}$ is unique, the gradient $\nabla_{\mathbf{w}_{\mathcal{A}}} = 0$ at $\hat{\mathbf{x}}^*_{\mathcal{A}}$.

Let $\mathbf{b}_j$ be any atom from $\mathbf{B}$ that is not in $\mathbf{B}_{\mathcal{A}}$ and which can be expressed as a linear combination of the atoms in $\mathbf{B}_{\mathcal{A}}$, *i.e.* $\mathbf{b}_j = \mathbf{B}_{\mathcal{A}}\mathbf{z}$ for some some non-zero vector $\mathbf{z}$. Let w_j be the weight assigned to $\mathbf{b}_j$ in the composition of $\mathbf{x}$. The derivative of $\mathrm{KL}(\mathbf{w})$ with respect to w_j at $\hat{\mathbf{x}}^*_{\mathcal{A}}$ is given by $\mathbf{z}^T\nabla_{\mathbf{w}_{\mathcal{A}}} = 0$. Consequently, $\mathbf{b}_j$ will not be chosen in the update step of the solution.

The rest of the proof follows by construction: at the initialization of the algorithm $\mathbf{B}_{\mathcal{A}}$ consists of only one atom. Subsequent increments of the active set only add newer atoms that are linearly independent of the current set, thereby maintaining linear independence of the atoms in the set.

REFERENCES

[1] Y.-C. Cho and S. Choi, "Nonnegative features of spectro-temporal sounds for classification," *Pattern Recognition Letters*, vol. 26, no. 9, 2005.

[2] N. Bertin, R. Badeau, and E. Vincent, "Enforcing harmonicity and smoothness in Bayesian non-negative matrix factorization applied to polyphonic music transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, 2010.

[3] J. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, 2011.

[4] T. Virtanen, "Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, 2007.

[5] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, "Non-negative matrix factorization based compensation of music for automatic speech recognition," in *Interspeech 2010*, Tokyo, Japan, 2010.

[6] D. Bansal, B. Raj, and P. Smaragdis, "Bandwidth expansion of narrow-band speech using non-negative matrix factorization," in *9th European Conference on Speech Communication and Technology*, Lisbon, Portugal, 2003.

[7] A. Ozerov, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, 2010.

[8] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio & music: from coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2009.

[9] J. Nikunen and T. Virtanen, "Object-based audio coding using non-negative matrix factorization for the spectrogram representation," in *Proceedings of the 128th Audio Engineering Society Convention*, London, UK, 2010.

[10] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proceedings of Neural Information Processing Systems*, Denver, USA, 2000, pp. 556–562.

[11] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations*. Wiley, 2009.

[12] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, 2007.

[13] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, 2004.

[14] J. F. Murray and K. Kreutz-Delgado, "Sparse image coding using learned overcomplete dicionaries," in *IEEE Workshop on Machine Learning for Signal Processing*, Sao Luis, Brazil, 2004.

[15] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, USA, 2003.

[16] N. B. C. Févotte and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. with application to music analysis," *Neural Computation*, vol. 21, no. 3, 2009.

[17] J. Carabias-Orti, F. Rodriguez-Serrano, P. Vera-Candeas, F. Canadas-Quesada, and N. Ruiz-Reyes, "Constrained non-negative sparse coding using learnt instrument templates for realtime music transcription," *Engineering Applications of Artificial Intelligence*, 2013, in press.

[18] F. Weninger and B. Schuller, "Optimization and parallelization of monaural source separation algorithms in the openBliSSART toolkit," *Journal of Signal Processing Systems*, vol. 69, no. 3, 2012.

[19] S. Bellavia, M. Macconi, and B. Morini, "An interior point Newton-like method for non-negative least-squares problems with degenerate solution," *Numerical Linear Algebra with Applications*, vol. 13, no. 10, 2006.

[20] D. Kim, S. Sra, and I. S. Dhillon, "Fast Newton-type methods for the least squares nonnegative matrix approximation problem," in *Proceedings of SIAM Conference on Data Mining*, Minneapolis, USA, 2007.

[21] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with constrained second-order optimization," *Signal Processing*, vol. 87, no. 8, 2007.

[22] C.-J. Hsieh and I. S. Dhillon, "Fast coordinate descent methods with variable selection for non-negative matrix factorization," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, San Diego, USA, 2011.

[23] H. Van hamme, "A diagonalized Newton algorithm for non-negative sparse coding," in *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Vancouver, Canada, 2013, accepted for publication.

[24] J. Kim and H. Park, "Fast nonnegative matrix factorization: An active-set-like method and comparisons," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, 2011.

[25] R. Zdunek and A. Cichocki, "Fast nonnegative matrix factorization algorithms using projected gradient approaches for large-scale problems," *Computational Intelligence and Neuroscience*, 2008, article ID 939567.

[26] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization," *Proceedings of the National Academy of Sciences USA*, vol. 100, no. 5, 2003.

[27] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, 2009.

[28] T. Blumensath, "Compressed sensing with nonlinear observations," *Preprint*, 2010.

[29] S. Bahmani, P. Boufounos, and B. Raj, "Greedy sparsity-constrained optimization," *Journal of Machine Learning Research*, 2013.

[30] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, 1993.

[31] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of Asilomar Conference on Signals, Systems and Computers*, 1993.

[32] D. Needell and J. A. Tropp, "Cosamp: iterative signal recovery from incomplete and inaccurate samples," *Communications of the ACM*, vol. 53, no. 12, 2010.

[33] P. Smaragdis, "Extraction of speech from mixture signals," in *Techniques for Noise Robustness in Automatic Speech Recognition*, T. Virtanen, R. Singh, and B. Raj, Eds. Wiley, 2012, pp. 87 – 108.

[34] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *Proceedings of the International Conference on Spoken Language Processing*, Pittsburgh, USA, 2006.

[35] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD and its non-negative variant for dictionary design," in *Proceedings of SPIE Conference on Wavelet Applications in Signal and Image Processing XI*, San Diego, USA, 2005.

[36] T. Virtanen and A. T. Cemgil, "Mixtures of gamma priors for non-negative matrix factorization based speech separation," in *Proceedings of the 8th International Conference on Independent Component Analysis and Blind Signal Separation*, Paraty, Brazil, 2009.

[37] J. F. Gemmeke and H. Van hamme, "An hierarchical exemplar-based sparse model of speech, with an application to ASR," in *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Honolulu, USA, 2011.

[38] P. Smaragdis, M. Shashanka, and B. Raj, "A sparse non-parametric approach for single channel separation of known sounds," in *Proceedings of Neural Information Processing Systems*, Vancouver, Canada, 2009.
[39] E. Vincent, N. Bertin, and R. Badeau, "Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription," in *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, Las Vegas, USA, 2008.
[40] A. T. Cemgil, "Bayesian inference for nonnegative matrix factorisation models," *Computational Intelligence and Neuroscience*, 2009, article ID 785152.
[41] A. Cichocki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92-A, no. 3, pp. 708–721, 2009.
[42] S. Boyd and L. Vanderberghe, *Convex Optimization*. Cambridge University Press, 2004.
[43] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.
[44] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *Journal of the Acoustical Society of America*, vol. 120, no. 5, 2006.
[45] M. Cooke, J. R. Hershey, and S. J. Rennie, "Monaural speech separation and recognition challenge," *Computer Speech and Language*, vol. 24, no. 1, 2010.
[46] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, October 1999.
[47] D. Sarkar, "Methods to speed up error back-propagation learning algorithms," *ACM Computing Surveys*, vol. 27, no. 4, 1995.
[48] J. R. Hershey, S. J. Rennie, and J. Le Roux, "Factorial models for noise robust speech recognition," in *Techniques for Noise Robustness in Automatic Speech Recognition*, T. Virtanen, R. Singh, and B. Raj, Eds. Wiley, 2012, pp. 311 – 345.
[49] J. O. Smith III, *Spectral audio signal processing*. on-line book, http://ccrma.stanford.edu/~jos/sasp/, 2007.
[50] J. F. Gemmeke and H. Van hamme, "Advances in noise robust digit recognition using hybrid exemplar-based techniques," in *Proceedings of Interspeech*, Portland, USA, 2012.

Tuomas Virtanen is an Academy Research Fellow and an adjunct professor in Department of Signal Processing, Tampere University of Technology (TUT), Finland. He received the M.Sc. and Doctor of Science degrees in information technology from TUT in 2001 and 2006, respectively. He has also been working as a research associate at Cambridge University Engineering Department, UK. He is known for his pioneering work on single-channel sound source separation using non-negative matrix factorization based techniques, and their application to noise-robust speech recognition, music content analysis and audio classification. In addition to the above topics, his research interests include content analysis of audio signals and machine learning.

Jort Florent Gemmeke is a postdoctoral researcher at the KU Leuven, Belgium. He received the M.Sc degree in physics from the Universiteit van Amsterdam (UvA) in 2005. In 2011, he received the Ph.D degree from the University of Nijmegen on the subject of noise robust ASR using missing data techniques. He is known for pioneering the field of exemplar-based noise robust ASR. His research interests are automatic speech recognition, source separation, noise robustness and acoustic modeling, in particular exemplar-based methods and methods using sparse representations.

Bhiksha Raj is an associate professor in the Language Technologies Institute of Carnegie Mellon University, with additional affiliations to the Machine Learning and Electrical and Computer Engineering departments of the University. Dr Raj completed is PhD from Carnegie Mellon University in 2000. From 2001-2008 he worked at Mitsubishi Electric Research Labs in Cambridge MA, where he led the research effort on speech processing. He has been at CMU since 2008. Dr Raj's research interests include speech and audio processing, automatic speech recognition, natural language processing and machine learning.