ACTIVE-SET NEWTON ALGORITHM FOR NON-NEGATIVE SPARSE CODING OF AUDIO

Tuomas Virtanen[⋆] *Bhiksha Raj*[†] *Jort F. Gemmeke*[‡] *Hugo Van hamme*[‡]

[⋆] Tampere University of Technology, Department of Signal Processing
[†]Carnegie Mellon University, Language Technologies Institute
[‡]KU Leuven, Department of Electrical Engineering

ABSTRACT

We propose a new algorithm to efficiently obtain non-negative sparse representations for audio. The spectrum of an audio signal is represented as a sparse linear combination of atoms taken from an overcomplete dictionary. The algorithm is based on minimizing the generalized Kullback-Leibler divergence between an observed magnitude spectrum and a non-negative linear combination of atoms, plus an ℓ_1 regularization term. The proposed method consists of an active-set method that iteratively updates a set of active atoms that have non-zero weights, using a Newton step where the weights of the active atoms are updated. The proposed method was evaluated using mixtures of two speakers, and it was shown to yield more than 10 times faster convergence in comparison to an established algorithm based on multiplicative update rules. Moreover, the ℓ_1 regularization was found to decrease the computation time and to improve the source separation performance.

Index Terms— sound source separation, non-negative matrix factorization, Newton algorithm, convex optimization, sparse coding

1. INTRODUCTION

Compositional models such as non-negative matrix factorization model non-negative data as a non-negative linear composition of non-negative components. The underlying metaphor of construction from parts makes them excellent models for many kinds of data, in particular audio, where they are used to characterize magnitude spectral representations of complex sounds as a non-negative linear composition of atomic spectral units. Compositional models of audio have been shown to have several applications in content analysis [1, 2], manipulation and enhancement [3, 4, 5], and coding [6, 7]. They have been found to be particularly useful for the analysis of sound mixtures, and the separation of mixtures into their constituents. The chief benefit of compositional models in these applications is their ability to represent the sound from any source as a composition of atomic sound units from that source. We will refer to these atomic sound units as "atoms", and to collections of atoms as a "dictionary". Mixtures of sounds become compositions of the atoms from the dictionaries of all the contributing sources, and separation of a mixture into its constituents simply becomes the problem of segregating the contributions of the atoms for the individual sources to the mixture.

For effective analysis or decomposition, however, two issues must be addressed. The first is the nature of the dictionary of atoms for any source. Although not the main topic of this paper, it is nevertheless useful to consider it briefly. Ideally, the dictionary for any source must capture all the sounds that can be generated by that source, and several supervised learning methods have been proposed for this purpose, which learn these dictionaries from example recordings of the source [8, 9]. Most natural sound sources are complex and can generate sounds with a large diversity of spectral characteristics. In order to effectively characterize the variety of sounds that they may produce, the dictionaries too must be large – larger dictionaries lead to a more accurate representation, and therefore better analysis or source separation results [10]. Frequently, such dictionaries become *overcomplete*, often greatly so, having many more entries that the dimensionality of the spectral vectors themselves.

The second issue, which is the topic of this paper, is that of determining *how* a given data (spectral) vector $\mathbf{x}$ must be decomposed in terms of a given dictionary matrix $\mathbf{B}$, *i.e.* how to determine the *weight* vector $\mathbf{w}$ such that $\mathbf{x} \approx \mathbf{Bw}$ most closely, and a specified *divergence* $D(\mathbf{x}, \mathbf{Bw})$ between $\mathbf{x}$ and $\mathbf{Bw}$ is minimized. Closed-form solutions do not exist for the minimization and iterative algorithms are required. The computational complexity of these algorithms increases with dictionary size, imposing constraints on the number of atoms. When the decomposition can be done offline, large dictionaries with tens of thousands of atoms may be used. If the decomposition must be real-time, dictionaries must be much smaller. In both cases, the number of atoms and, consequently, the accuracy of the representation must be compromised in order to make the computations feasible. There is thus a need for efficient algorithms for the decomposition, particularly for large dictionaries.

Additional issues must also be considered. The actual algorithm will depend on this divergence $D(\mathbf{x}, \mathbf{Bw})$ that is minimized. For many types of data the most popular divergence is the squared error, but for audio applications other divergences such as the generalized Kullback-Leibler (KL) divergence have been found to be more suitable [3, 11, 12]. We will use the KL divergence in this paper.

Another issue to be considered is *sparsity*. When the dictionary is overcomplete, the relation $\mathbf{x} \approx \mathbf{Bw}$ is underspecified, and an uncountable multiplicity of equivalent solutions may exist for the decomposition. To obtain structurally meaningful solutions additional constraints must be enforced. The most common constraint is that of sparsity: the weight vector $\mathbf{w}$ is required to be sparse, *i.e.*, that most of its components be zero-valued. Intuitively, the requirement for sparsity implies that although the source itself is capable of generating a large variety of spectral structures, as embodied by atoms in the dictionary, only a small number of these are present in any *given* spectral vector. When sparsity constraints are thus employed, the decomposition is often referred to as *non-negative sparse coding*, a term we too will use. The requirement of sparsity by itself is not specific to overcomplete decompositions – sparsity may be enforced even when the dictionaries are not overcomplete. Sparsity is usually imposed through the addition of an ℓ_1 regularization term $\lambda||\mathbf{w}||_1$ to

T. Virtanen has been funded by the Academy of Finland grant number 258708. The research of Jort F. Gemmeke was funded by IWT-SBO project ALADIN contract 100049

the divergence that is minimized: minimization of the ℓ_1 norm of the $\mathbf{w}$ also naturally biases it towards sparsity [13].

The most commonly used decomposition algorithms [14] are based on first-order optimization. These are easy to implement, accommodate a variety of divergences, and can also be extended to more complex models. Second-order algorithms are known to be generally more efficient than first-order methods and a number have been proposed. Several of these are constrained to *compact* representations where the number of atoms is smaller then the dimensionality of the spectral vectors [15, 16, 17]. Other second-order methods are specific to minimizing the squared error divergence [18, 19], which, as we mentioned earlier, is suboptimal for audio analysis applications. Yet others minimize a larger class of divergences [20, 21], however they approximate the Hessian of the objective function to be minimized by a diagonal matrix, thereby losing some of its structure.

In prior work [10] we showed that non-negative representations are naturally sparse, *i.e.*, that when appropriately computed, the estimated weight vectors will have only a small number of non-zero entries even when sparsity is not explicitly imposed. Based on this we proposed an active-set method, dubbed ASNA, that employs a Newton algorithm using a full Hessian matrix for optimization. The proposed algorithm was shown to be very efficient at computing sparse non-negative decompositions even when explicit sparsity constraints are not applied.

In this paper we extend the method in [10] to explicitly introduce sparsity constraints. We do so by introducing an ℓ_1 regularization to the active-set Newton method, and show that it is able to achieve further improved decompositions, as evaluated by a signal-separation task, without loss of efficiency.

The rest of this paper is as follows. In Section 2 of the paper we describe the model and the criterion for parameter estimation. Section 3 describes the proposed algorithm, and Section 4 evaluates its performance in comparison to other available algorithms.

2. NON-NEGATIVE SPARSE CODING OF AUDIO

In non-negative sparse coding of audio, an $F \times 1$ observation vector $\mathbf{x}$ is modeled as a weighted linear combination of atom vectors $\mathbf{b}_n$ from a *dictionary* as

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

where w_n is the non-negative weight applied to $\mathbf{b}_n$, and N is the number of atoms in the dictionary. We rewrite the model compactly using a matrix-vector product as

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

where the $F \times N$ dictionary matrix is defined as $\mathbf{B} = \begin{bmatrix}\mathbf{b}_1, \dots, \mathbf{b}_N\end{bmatrix}$ and an $N \times 1$ weight vector is defined as $\mathbf{w} = \begin{bmatrix}w_1, \dots, w_N\end{bmatrix}^T$.

In audio applications, the observation vector and atoms typically represent short-time magnitude (square root of power) or power spectra of audio, but the proposed method is not constrained to any specific representation, only that the observations and atoms are non-negative.

Our objective is to estimate the weights $\mathbf{w}$, given the observation vector $\mathbf{x}$ and the dictionary $\mathbf{B}$. The weights are estimated by minimizing the cost function

$$f(\mathbf{w}) = \mathrm{KL}(\mathbf{x}||\mathbf{B}\mathbf{w}) + \lambda||\mathbf{w}||_1 \tag{3}$$

where $\mathrm{KL}(\mathbf{x}||\mathbf{B}\mathbf{w})$ is the Kullback-Leibler (KL) divergence between the observation vector $\mathbf{x}$ and the model $\mathbf{B}\mathbf{w}$, $||\mathbf{w}||_1$ is the ℓ_1 norm of the weight vector, and λ is a sparseness parameter. The KL divergence is defined as

$$\mathrm{KL}(\mathbf{x}||\hat{\mathbf{x}}) = \sum_i d(x_i, \hat{x}_i), \tag{4}$$

where function d is defined as

$$d(p,q) = \begin{cases} p\log(p/q) - p + q & p > 0 \ \text{ and } \ q > 0 \\ q & p = 0 \\ \infty & p > 0 \ \text{ and } \ q = 0. \end{cases} \tag{5}$$

3. PROPOSED ACTIVE-SET NEWTON ALGORITHM

We have shown [10] that even without explicit sparseness constraints, the minimizers of (3) are sparse, meaning that only a small number of weights are non-zero. Based on this observation, we can calculate (2) more efficiently by explicitly keeping track of a set $\mathcal{A}$ of active atoms. We write:

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

The active set is iteratively updated as described below to find the optimal set of active atoms. For estimating the weights in the active set, we use the Newton algorithm. The proposed algorithm consists of the following steps:

Step 1: The active set is initialized with the single atom $\mathbf{b}_n$ and its weight w_n that together give the minimum cost $\mathrm{KL}(\mathbf{x}||w_n\mathbf{b}_n) + \lambda w_n$. This is done as follows. The optimal weight for any individual atom $\mathbf{b}_n$ is obtained by setting the derivative of the above cost with respect to weight w_n to zero and solving for w_n, which gives

$$w_n = \frac{\mathbf{1}^T\mathbf{x}}{\mathbf{1}^T\mathbf{b}_n + \lambda}, \tag{7}$$

where $\mathbf{1}$ is a all-one vector of length F. The optimal weight and the corresponding cost $\mathrm{KL}(\mathbf{x}||w_n\mathbf{b}_n) + \lambda w_n$ are evaluated for each atom $\mathbf{b}_n$, $n = 1...N$. The atom giving the lowest cost and its corresponding weight are used to initialize the active set.

Step 2: The active set is updated by adding an atom having the most negative partial derivative of the cost function with respect to weight w_n. The partial derivative is given as

$$\frac{\partial}{\partial w_n} f(\mathbf{w}) = \mathbf{b}_n^T(\mathbf{1} - \frac{\mathbf{x}}{\mathbf{B}\mathbf{w}}) + \lambda. \tag{8}$$

The weight that will be added to the active set is initialized to a small positive value (10^{-15} in our implementation). If all the weight derivatives of (8) are positive, adding a new atom to the active set will not decrease the cost, in which case no new atom is added.

Step 3: The weights of the atoms in the active set are updated using the Newton method as

$$\mathbf{w}_{\mathcal{A}} \leftarrow \mathbf{w}_{\mathcal{A}} - \alpha \mathbf{H}_{\mathbf{w}_{\mathcal{A}}}^{-1} \nabla_{\mathbf{w}_{\mathcal{A}}}. \tag{9}$$

where $\mathbf{w}_{\mathcal{A}}$ denotes the vector of weights of the atoms in the active set, and $\mathbf{H}_{\mathbf{w}_{\mathcal{A}}}$ is the Hessian matrix of the cost function with respect to the active-set weight vector. We will also denote the dictionary matrix consisting of the active atoms as $\mathbf{B}_{\mathcal{A}}$. The gradient of the cost function with respect to the active-set weight vector is given as

$$\nabla_{\mathbf{w}_{\mathcal{A}}} = \mathbf{B}_{\mathcal{A}}^T(\mathbf{1} - \frac{\mathbf{x}}{\mathbf{B}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}}}) + \lambda, \tag{10}$$

and the Hessian $\mathbf{H}_{\mathbf{w}_{\mathcal{A}}}$ is

$$\mathbf{H}_{\mathbf{w}_{\mathcal{A}}} = \mathbf{B}_{\mathcal{A}}^T \ \mathrm{diag}(\frac{\mathbf{x}}{(\mathbf{B}_{\mathcal{A}}\mathbf{w}_{\mathcal{A}})^2}) \ \mathbf{B}_{\mathcal{A}}. \tag{11}$$

In (9) α is a step size that is calculated as $\alpha = \min_{r_i>0} r_i$, where $\mathbf{r} = \mathbf{w}_{\mathcal{A}}/(\mathbf{H}^{-1}_{\mathbf{w}_{\mathcal{A}}}\nabla_{\mathbf{w}_{\mathcal{A}}})$. The maximum value of the step size is limited to 1, which corresponds to the standard Newton algorithm. The above step size calculation guarantees the resulting weights to be non-negative. As a result of the above Newton update, an entry in the weight vector may get a zero value. In this case the atom is removed from the active set. In order to ensure the numerical stability of the inversion in (9), we apply Tikhonov regularization, i.e., add an identity matrix multiplied by small positive constant (10^{-10} in our implementation) to the Hessian matrix $\mathbf{H}_{\mathbf{w}_{\mathcal{A}}}$ prior to taking the inverse.

The algorithm is iterated by repeating steps 2 and 3 until all the weight derivatives in (8) are non-negative (in practice we use a lower threshold -10^{-15}), or the norm of the gradient in (10) is zero (in practice we use an upper threshold 10^{-15}). The above stopping criteria are met only when the global optimum has been found [10], because $f(\mathbf{w})$ is a convex function: the KL divergence is convex [10], the ℓ_1 norm of weights is convex, the sum of two convex functions is convex, and the non-negative orthant which is the feasible region of the weights is convex. As in [10], two Newton updates (step 3) are done before each update of the active set (step 2).

4. EVALUATION

We evaluate the proposed method from three different angles: the effect of sparsity on the convergence of the algorithm, the actual value of the objective function achieved in comparison to other methods, and performance on a source separation task that depends on effective non-negative sparse coding.

As acoustic material we use synthesized mixtures of two speakers. We use the subset of the GRID corpus [22] that was used as the training set in the Speech Separation Challenge [23]. The material consists of 34 speakers, each uttering 500 short sentences having a small vocabulary. The sampling frequency of the signals is 25 kHz. The test set consists of 100 signals, each of which is a mixture of two utterances, both randomly selected from a random speaker. The root-mean-square levels of the speakers in the mixtures were equal.

The signals were represented using their short-time magnitude spectra. A 60-ms Hanning window with 15-ms frame hop was used to window the signals into a sequence of frames, and the absolute value of the discrete Fourier transform was used to calculate the observation vector within each frame. Only positive frequencies were used, which leads to 751 frequency bins.

A speaker-dependent dictionary of atoms was generated for each speaker using all the utterances not in the test set as training material. The training material was represented similarly to the test data using their short-time magnitude spectra. The observation vectors in the training data were clustered by minimizing the KL divergence between cluster centers and observations within a cluster, as in [10]. The cluster centers were normalized to unit ℓ_2 norm and used as atoms in a dictionary. Three different dictionary sizes per speaker were considered: 50, 500, and 5 000 atoms.

When representing a mixture signal, the dictionaries of the speakers within the mixture (using ground truth information about the speaker identities) were concatenated to form dictionaries of sizes 100, 1 000, and 10 000 atoms. The training and test sets, dictionaries, and evaluation procedure were the same as in [10].

We first examine the effect of the sparseness parameter λ on the convergence of the proposed method, and the effect of λ on the size of the obtained set of active atoms. The proposed method is applied on all the test mixtures and the size of the active set after the proposed method has converged and the number of iterations required for convergence is calculated. Values 0, 0.01, 0.1, 1, and 10 of λ were tested. Figure 1 illustrates the averages of the above metrics as a function λ. Increasing λ decreases both the active set size and the number of iterations, and there is apparently a high correlation between the metrics. Since the proposed method is stopped only after it has reached the global minimum of the objective function, the results show that the method is able to converge to the global optimum with a limited number of iterations even when $\lambda > 0$ is used. Actually, higher values of λ make the proposed method converge faster. The sparseness regularization $\lambda > 0$ affects larger dictionaries more, decreasing the size of the active set as λ increases.

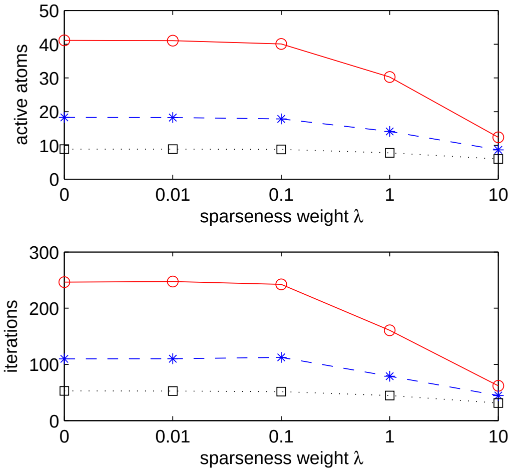

Fig. 1. Number of active atoms in the optimal active set (upper panel) and the number of iterations required for the proposed method to converge as a function of the sparseness weight λ for different dictionary sizes. Each line in both panels correspond to dictionary sizes 10 000, 1 000, and 100 atoms (from top to down).

Second, we compare the convergence speed of the proposed active-set Newton method, ASNA, to two other successful algorithms: the popular multiplicative update rules proposed in [14] augmented with the sparseness term λ [8] (here referred to as EM), and an alternate Newton method [21] which uses a diagonal approximation of the Hessian for the stationary-point condition of the objective, known as DNA. All the algorithms were implemented using MATLAB and a reasonable effort was made to use computationally efficient matrix-vector operations and to avoid loops. Implementations of ASNA and DNA are available on-line at `http://www.cs.tut.fi/~tuomasv/software.html` and `http://www.esat.kuleuven.be/psi/spraak/downloads/`.

As an evaluation metric we use the value of objective function $f(\mathbf{w})$ as the function of the computation time. The objective function is convex; as a result all the algorithms will eventually converge to the same value of the objective function, albeit at different speeds. Instead of a theoretical analysis of computational complexity we measure the used CPU time after each iteration since it gives a more informative picture of the performance in practical scenarios. The global minimum of objective function $f(\mathbf{w})$ is different for each observation, dictionary size, and value of λ, which makes the comparison of results difficult. As an evaluation metric we therefore calculate a normalized KL divergence by subtracting the globally optimal value of the cost function, which is obtained by executing ASNA until it converges. The average normalized KL divergence is

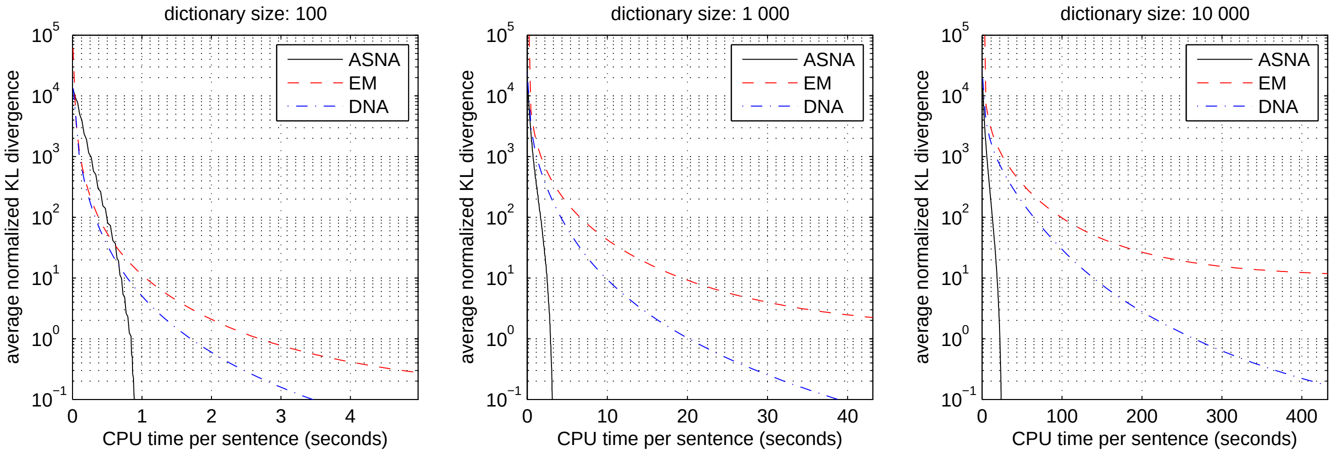

Fig. 2. The average normalized KL divergence per sentence as the function of the cumulative CPU time.

calculated by averaging over all the frames and test signals.

Figure 2 illustrates the average normalized KL divergence as the function of used CPU time for different dictionary sizes. Here, we use $\lambda = 1$, which was found to produce the best source separation results (shown later). For all the dictionary sizes, ASNA is able to reach asymptotically faster convergence in comparison to the reference algorithms. Especially on two larger dictionary sizes (1 000 and 10 000 atoms) ASNA reached the lowest normalized KL divergence obtained with the established EM algorithm more than 10 times faster. When comparing the results to the case $\lambda = 0$ for which the results were presented in [10], we observe that ASNA becomes more efficient in comparison to EM when $\lambda = 1$ is used.

Third, we evaluate the effect of λ on the source separation performance. Sparseness has been found beneficial in source separation [24, 25], but there are also studies where sparseness has been reported to not increase separation quality [3]. The difference between the above results is explained by differences in the separation framework (unsupervised vs. supervised), acoustic material (speech vs. music), and dictionary type (parts-based vs. clustered).

Before we proceed, it is worth summarizing the separation task: we are given magnitude spectral vectors $\mathbf{x}_z$ for a mixed signal $Z = X + Y$. We assume that any magnitude spectrum $\mathbf{x}_z$ for Z can be well approximated as $\mathbf{x}_z = \mathbf{x}_x + \mathbf{x}_y$, where $\mathbf{x}_x$ and $\mathbf{x}_y$ are magnitude spectral vectors for the constituent signals X and Y that compose Z. The objective is to recover $\mathbf{x}_x$ and $\mathbf{x}_y$ from $\mathbf{x}_z$. We will also assume that we are in possession of dictionaries $\mathbf{B}_x$ and $\mathbf{B}_y$ for the sources that generated X and Y. The algorithm for separation is now simple: we create $\mathbf{B}_z = [\mathbf{B}_x \mathbf{B}_y]$. We then decompose $\mathbf{x}_z = \mathbf{B}_z \mathbf{w}_z$ using the decomposition algorithm to be evaluated. The estimated weight vector $\mathbf{w}_z$ can be written as $\mathbf{w}_z = [\mathbf{w}_x^\top \mathbf{w}_y^\top]^\top$. The separated estimate for $\mathbf{x}_x$ is now obtained by "Wiener-style" reconstruction as $\hat{\mathbf{x}}_x = \frac{\mathbf{B}_x \mathbf{w}_x}{\mathbf{B}_z \mathbf{w}_z}$. The separated estimate for $\mathbf{x}_y$ is obtained in the same fashion [9]. The separated sequence of spectral vectors is reverted to a signal as in [9].

We evaluate the actual separation performance through the signal-to-distortion (SDR) ratio, which is calculated for each of the speakers, in each test signal, and subsequently averaged over both speakers and all test signals. We show the results only for ASNA. For EM and DNA the SDR results are almost identical.

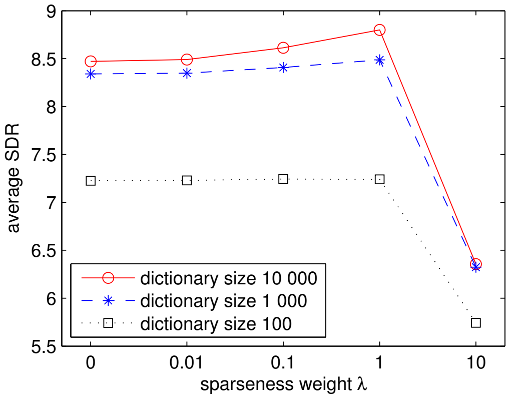

Fig. 3. Average SDRs of the separated speech as the function of the sparseness weight λ for different dictionary sizes.

The average SDR for different dictionary sizes as a function of the sparseness weight λ is illustrated in Figure 3. For large dictionary sizes having $\lambda > 0$ is found beneficial. At all dictionary sizes a too large λ decreases the SDR. The larger the dictionary size, the bigger the benefit of using appropriate λ. In practical usage scenarios, an optimal value of λ can be chosen by using development material that matches better the usage scenario.

5. CONCLUSIONS

We have proposed a novel ASNA algorithm to efficiently obtain non-negative sparse representations for audio. The algorithm is based on minimizing the generalized Kullback-Leibler divergence between an observed magnitude spectrum and a non-negative linear combination of atoms, plus an ℓ_1 regularization term. The proposed method consists of an active-set method that iteratively updates a set of active atoms that have non-zero weights, using a Newton step where the weights of the active atoms are updated. The proposed ASNA method was evaluated using mixtures of two speakers, and it was shown to yield more than 10 times faster convergence in comparison to an established algorithm based on multiplicative update rules. Moreover, the ℓ_1 regularization was found to improve the source separation performance.

6. REFERENCES

[1] Y.-C. Cho and S. Choi, “Nonnegative features of spectro-temporal sounds for classification,” *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1327 – 1336, 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, pp. 538 – 549, 2010.

[3] 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, pp. 1066 – 1074, 2007.

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

[5] A. Ozerov and C. Févotte, “Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 550 – 563, 2010.

[6] 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.

[7] 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.

[8] 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, pp. 2067 – 2080, 2011.

[9] B. Raj, T. Virtanen, S. Chaudhure, and R. Singh, “Non-negative matrix factorization based compensation of music for automatic speech recognition,” in *Proceedings of Interspeech 2010*, Tokyo, Japan, 2010.

[10] T. Virtanen, J. Gemmeke, and B. Raj, “Active-set Newton algorithm for overcomplete non-negative representations of audio,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 11, 2013.

[11] 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*, pp. 1671 – 1680, 2013.

[12] 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, pp. 267 – 277, 2012.

[13] D. L. Donoho, “For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[14] 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.

[15] 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.

[16] 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.

[17] R. Zdunek and A. Cichocki, “Nonnegative matrix factorization with constrained second-order optimization,” *Signal Processing*, vol. 87, no. 8, pp. 1904 – 1916, 2007.

[18] 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, pp. 3261 – 3281, 2011.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] M. Cooke, J. R. Hershey, and S. J. Rennie, “Monaural speech-separation and recognition challenge,” *Computer Speech and Language*, vol. 24, no. 1, 2010.

[24] 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.

[25] 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.