Binary non-negative matrix deconvolution for audio dictionary learning

Szymon Drgas, Tuomas Virtanen, Jörg Lücke, Antti Hurmalainen

***Abstract*—In this study we propose an unsupervised method for dictionary learning in audio signals. The new method, called binary non-negative matrix deconvolution (BNMD), is developed and used to discover patterns from magnitude-scale spectrograms. The BNMD models an audio spectrogram as a sum of delayed patterns having binary gains (activations). Only small subsets of patterns can be active for a given spectrogram excerpt. The proposed method was applied to speaker identification and separation tasks. The experimental results show that dictionaries obtained by the BNMD bring much higher speaker identification accuracies averaged over a range of SNRs from -6 dB to 9 dB (91.3%) than the NMD-based dictionaries (37.8–75.4%). The BNMD also gives a benefit over dictionaries obtained using vector quantization (87.8%). For bigger dictionaries the difference between the BNMD and the VQ is getting smaller. For the speech separation task the BNMD dictionary gave a slight improvement over the VQ.**

***Index Terms*—Sparse coding, speaker recognition, speech separation**

I. Introduction

Many classes of audio signals can be treated as a composition of repeating acoustic events. In the case of music these acoustic events are notes, in the case of environmental noise these can be sounds specific for various sources (for example in a street it can be a passing car), for speech signals recurring acoustic events are for example phones. Representing audio recordings as a combination of atoms has been used to obtain state-of-the-art results in many domains. Application of such techniques for speech separation was reported e.g. in [1], [2]. This kind of techniques has also been employed in speech/speaker recognition systems [3], [4]. In the second CHiME speech separation and recognition challenge [5], in the best performing system in the Track 1 [6] non-negative matrix factorization was used to separate a speech signal from unwanted sounds. NMF in this case was used to represent a noisy signal as a combination of exemplars sampled directly from training data. Another system that gave good results in the CHiME challenge was also based on exemplars with NMF, but the recognition was done in a different way [7]. Other examples of applications where dictionaries are used in state-of-the-art solutions, are automatic music transcription [8] or instrument recognition [9].

The discovery of atoms that reflect repeating sounds can be done by means of unsupervised learning methods. One of the simplest approaches obtaining a collection of atoms that represent acoustic events is to sample spectra or spectro-temporal patches. For large sets of atoms, this method can lead to good results for the speech recognition task [10]. A possibility to reduce the number of atoms and to obtain possibly more meaningful patterns is to use a clustering method (for example k-means [11]).

In the k-means algorithm, each cluster is represented by its mean. The means obtained by this algorithm form the set of atoms which is in this context often called a 'codebook'. Codebook learning using k-means can be described as finding the best set of atoms so that each observation (spectrum or spectro-temporal excerpt) is well approximated by one of the atoms. This approach can be generalized by requiring that each of the observations is well approximated by a linear combination of atoms. The set of learned atoms is in this context often called a 'dictionary' (while dictionary learning also sometimes includes codebook learning as special case). A dictionary learning approach is called 'sparse' if only few atoms (on average) are of significance for each linear reconstruction. Typically, sparsity refers to reconstruction weights with high values for some atoms and values close to zero for most others. If the coefficients for most atoms are exactly zero, we speak of 'hard sparsity' [12], [13], [14]. The problem of finding sparse representations (i.e., sparse dictionaries) is computationally intractable in general. Therefore, a number of approximations have been developed ranging from deterministic approaches such as matching pursuit [15], orthogonal matching pursuit [16], K-SVD, ICA [17], [18] to probabilistic approaches such as sparse coding [19] and its different versions.

A desirable property for spectrogram representations is to allow only constructive combinations of atoms. By a constructive combination we mean a combination in which its component can be only added [20]. It (on average)is obtained by constraining values of atoms and their activations to be non-negative. Mathematical models that characterize data by constructive combination are named compositional. A popular compositional model is non-negative matrix factorization [21]. Non-negative matrix factorization is an algorithm in which a dictionary can be learned for a representation of spectra. An example of using NMF to learn a dictionary can be found, e.g., in [22]. In addition to the originally deterministic definitions of NMF, probabilistic version have become popular in recent years [23], [12]. In this article we deal with models that are both compositional and sparse. An example of non-negative sparse representation with l_1 norm-based sparse coding is

S. Drgas is with the Chair of Control and Systems Engineering, Poznan University of Technology, e-mail: szymon.drgas@put.poznan.pl

T. Virtanen is with the Department of Signal Processing, Tampere University of Technology, e-mail fistname.lastname@tut.fi

J. Lücke is with the Cluster of Excellence Hearing4all and the School of Medicine and Health Sciences, University of Oldenburg, Oldenburg, Germany, e-mail: joerg.luecke@uol.de

A. Hurmalainen is with Yousician Ltd., e-mail: antti@yousician.com

Manuscript received April 19, 2005; revised September 17, 2014.

sparse non-negative matrix factorization (sparse NMF) [24].

In dictionary-based methods an observation is represented as a combination of atoms. Atoms may correspond to its parts. For example, a spectrogram excerpt can be the observation, while its parts can correspond to acoustic events. This representation can be potentially more robust than clustering approaches, as distortion of one part is only partially detrimental for interpretation of the observation. However, dictionary learning including NMF may for our purposes have the notable downside of separately representing units of speech both spectrally and temporally into multiple atoms. For example, a phone that is characterized by a particular structure of formants may become split into several single-formant atoms, which are also activated during other speech or noise events. Consequently, the separation and classification capability of such atoms is lower than for atoms which model their corresponding speech patterns as a whole [3, p. 27]. Models using 'hard sparsity' have in this context been found to be comparably high performing for classification [25], which argues for improved discriminative power of their learned atoms. An interesting model for NMF is therefore binary NMF (BNMF) [12] which enforces hard sparsity for compositional and non-negative sparse dictionaries. In addition to potentially improved atoms, e.g., for classification, it is straight-forward to interpret the sparsity parameter of binary models as the average number of activated (non-zero) atoms (in contrast to sparsity imposed using an l_1 penalty term for instance). This makes it possible to constrain the number of activated atoms in a specific way, or to potentially learn the sparsity from data.

For the purposes of this work, the basic use of the sparse representation techniques is to learn a representation of short time spectra that is composed of atoms. As mentioned earlier, sparse representations can be also used to represent spectrogram excerpts. Dictionary learning for longer spectrogram excerpts may result in atoms containing spectro-temporal patterns characteristic to sound sources. Modeling of such spectro-temporal patterns can be done by using a sliding window approach [26]. Frames of a spectrogram are windowed and concatenated. For each shift of the sliding window, sparse representation is obtained independently. In methods known from the literature, window lengths between 100 and 500 ms are used [26], [10]. In this case, the window spans a phoneme or a whole syllable. Another possibility is to use a convolutive model, where a modeled spectrogram is represented as a combination of temporally shifted atoms that contain spectro-temporal patterns. In [27] and independently in [28], non-negative matrix deconvolution (also known as convolutive NMF) was introduced. It was shown that atoms learned using this method have phoneme-like structure. Similarly in [29], it was shown that sparsity of activations helps to learn phonemes.

In this study, a novel binary non-negative matrix deconvolution (BNMD) is proposed to model a spectrogram. This model is similar to non-negative matrix deconvolution but the elements of the activation matrix can have binary values only. It can be applied to spectrograms in magnitude-scale, and it uses linear superposition to model a mixture of sounds coming from several sources. The information contained in the extracted components is evaluated in terms of speaker-dependent information. Instead of a greedy method or l_1-regularized sparse coding, we apply truncated EM approximations to train a probabilistic data model. Truncated approximations have previously been used for dictionary learning and are particularly well suited for models with hard sparsity [12], [13], [14]. For our purposes, we develop a truncated approximation for BNMD assuming a Poisson distribution as noise model, which can be interpreted as using KL-divergence as an error measure of fitting the model to the data (compare [21]). The proposed method is compared with sparse NMD and vector quantization. In vector quantization method, observations are obtained using a sliding window, without taking into account different temporal alignments of data within a window. Accurate representation using such model requires one centroid to represent each temporal alignment of data, which makes it in- efficient, and results in temporally blurred centroids.

This article is structured as follows: In Section II, dictionary learning using linear models with sparsity imposing criteria are discussed in detail. Next, in Section III, the BNMF algorithm which is the basis of the proposed BNMD is described. This is followed by the presentation of the BNMD method. In Section VI, experimental results are shown. Finally, conclusions are enumerated in Section VII.

II. Non-negative matrix deconvolution

Linear non-negative models which use atoms spanning multiple frames, model magnitude spectrogram $\mathbf{Y} \in \mathbb{R}_{\geq 0}^{D \times N}$, where D is a number of frequency bands and N is a number of spectrograms' frames, as a linear combination of temporally shifted atoms as

$$\mathbf{Y} \approx \sum_{t=1}^{T} \mathbf{W}(t) \overset{t-1\rightarrow}{\mathbf{S}} , \tag{1}$$

where $\mathbf{W}(t) \in \mathbb{R}_{\geq 0}^{D \times H}$ is a dictionary matrix with non-negative entries. The dictionary contains H atoms. Variable $\mathbf{S} \in \mathbb{R}_{\geq 0}^{H \times N}$ denotes an activation matrix. The dictionary dependence on t means that there is a temporal structure in the atoms. This means that each atom $\mathbf{W}_h = [\mathbf{w}_h(1) \ \dots \ \mathbf{w}_h(T)]$, where $\mathbf{w}_h(t)$ is the h'th column of $\mathbf{W}(t)$, corresponds to a $D \times T$ spectro-temporal patch spanning T frames of a spectrogram. An operator over the activation matrix $\overset{t\rightarrow}{\mathbf{S}}$ denotes the shift of each column of matrix $\mathbf{S}$ to the right. The t first columns are padded with zeros. For example:

$$\mathbf{S} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \qquad \overset{2\rightarrow}{\mathbf{S}} = \begin{bmatrix} 0 & 0 & 1 & 2 \\ 0 & 0 & 5 & 6 \end{bmatrix} . \tag{2}$$

The parameters of this model, namely matrices $\mathbf{W}(t)$ and $\mathbf{S}$ can be estimated by minimizing the Kullback-Leibler divergence

$$\begin{aligned} \min_{\mathbf{W}(1),\dots,\mathbf{W}(T),\mathbf{S}} \quad & KL(\mathbf{Y} \| \sum_{t=1}^{T} \mathbf{W}(t) \overset{t-1\rightarrow}{\mathbf{S}}) + \lambda \|\mathbf{S}\|_1 \\ \text{s.t.} \quad & \mathbf{W}(t) \geq \mathbf{0} \\ & \operatorname{diag}(\mathbf{W}^{\mathrm{T}}\mathbf{W}) = \mathbf{1} \\ & \mathbf{S} \geq \mathbf{0} \, , \end{aligned} \tag{3}$$

where $KL(\cdot\|\cdot)$ is the Kullback-Leibler divergence defined as

$$KL(\mathbf{A}\|\mathbf{B}) = \sum_i \sum_j A_{ij} \log \frac{A_{ij}}{B_{ij}} , \tag{4}$$

matrix $\mathbf{W}$ is defined as

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}(1) \\ \vdots \\ \mathbf{W}(T) \end{bmatrix} , \tag{5}$$

diag$(\cdot)$ is a vector that contains diagonal elements of a given matrix, and $\mathbf{1}$ is a vector with all elements equal to 1. As in sparse modeling a small number of active components is desired, l_1 norm ($\|\cdot\|_1$) is used to impose sparsity on the activation matrix by adding a term $\lambda\|\mathbf{S}\|_1$, where λ is the regularization parameter. In [23] and [30], it was shown that NMF can be interpreted as a generative model where $\mathbf{W}(t)$ and $\mathbf{S}$ are deterministic parameters. The modeled spectrogram is treated as a set of random variables Y_{dn}

$$Y_{dn} = \sum_{t=1}^{T} \sum_{h=1}^{H} C_{dnth} \, . \tag{6}$$

Latent variables in the model are C_{dnth} with the Poisson prior distribution

$$C_{dnth} \sim \mathcal{P}(C_{dnth}; W_{dh}(t) S_{h,n-t+1}) , \tag{7}$$

where $\mathcal{P}(k;\lambda) = \frac{\lambda^k \exp(-\lambda)}{k!}$. Because of the superposition property of the Poisson distribution, Y_{dn} has the Poisson distribution

$$Y_{dn} \sim \mathcal{P}\left(Y_{dn}; \sum_{t=1}^{T} \sum_{h=1}^{H} C_{dnth} \right) . \tag{8}$$

The parameters of the generative model can be estimated using the expectation-maximization (EM) algorithm, by iterating expectation (E) and maximization (M) steps. The M-step update expressions can be derived by equating partial derivatives of the free energy function

$$\mathcal{F} = \sum_{C_{dnth}} q(C_{dnth}) \log p(\mathbf{Y}, \mathbf{C} | \mathbf{W}(t), \mathbf{S}) + \mathcal{H}[q] \tag{9}$$

to zero with respect to the optimized parameters. Function $q(\cdot)$ is an approximation of posterior distribution of latent variables C_{dnth} (whose parameters are estimated during E-step), while $\mathcal{H}[q]$ is its Shannon entropy. In this case, M-step update formulas are exactly the same as formulas derived by Lee and Seung [21]. Both formulas depend on the sufficient statistics of the posterior distribution of C_{dnth} which is multinomial.

The NMD with l_1 sparsity term can be interpreted as a model specified by Equations (6) and (7), with the difference that S_{hn} denotes latent variables distributed according to the gamma distribution with the shape parameter equal to 1 [31]. Although this unimodal, decaying distribution leads to very good results in many applications, substantial part of mass of the prior distribution of activations is for small values of activations. It is not realistic in the case of atoms representing recurring speech patterns. In this case, most of atoms are not active in a given observation. There is a small number of active components that have non-zero values of the activations. An example of a distribution that can be more suitable is the Bernoulli distribution of the binary random variables. An example of a model with latent binary random variables is BNMF [12], which is described in the following section. Additionally, there is no direct relation between the sparsity parameters and the number of active atoms in a given observation. In the case of models with binary activations, the sparsity parameter is directly related to the expected number of active atoms.

III. Binary non-negative matrix factorization

The expectation-truncation NMF [12], which is referred to as binary non-negative matrix factorization (BNMF), is a probabilistic model in which an observation is treated as a linear combination of atoms. In contrast to NMF, the activations are binary and distributed according to the Bernoulli distribution. BNMF thus models the presence or absence of a component (dictionary element) rather than its intensity. Priors enforcing exact 'hard' zeros were in other studies found to result in better features for classification [25], which motivates the use of binary priors for this study. The distribution of activations s_h can be written as

$$S_{hn} \sim \mathcal{B}(S_{hn}, \pi) , \tag{10}$$

$$p(\mathbf{s}_n) = \prod_{h=1}^{H} \pi^{S_{hn}} (1-\pi)^{1-S_{hn}} , \tag{11}$$

where $\mathbf{s}_n = [S_{1n} \ \ldots \ S_{Hn}]^{\mathrm{T}}$ and π is the parameter of the Bernoulli distribution. In [12], this parameter is the same for all elements of vector $\mathbf{s}_n$. Parameter π can be fixed [32], [12], optimized with respect to a task or updated during M-step [13], [33]. The observed variables Y_{dn} are normally distributed around $\sum_h W_{dh} S_{hn}$:

$$p(\mathbf{y}_n | \mathbf{s}_n) = \prod_{d=1}^{D} \mathcal{N}(Y_{dn}; \sum_{h=1}^{H} W_{dh} S_{hn}, \sigma^2) , \tag{12}$$

where $\mathbf{y}_n = [Y_{1n} \ \ldots \ Y_{Dn}]^{\mathrm{T}}$, W_{dh} is an element of matrix $\mathbf{W}$ whose columns are atoms, and variance σ^2 is a parameter of the model. In order to obtain parameter values (W_{dh} and σ), the log-likelihood of data

$$\mathcal{L} = \sum_{n=1}^{N} \log p(\mathbf{y}_n | \mathbf{W}, \sigma) = \sum_{n=1}^{N} \log \sum_{\mathbf{s}_n \in \{0,1\}^H} p(\mathbf{y}_n, \mathbf{s}_n | \mathbf{W}, \sigma) \tag{13}$$

should be maximized. In order to maximize the log-likelihood, the expectation-maximization approach can be applied i.e. the free-energy function is maximized instead of the log-likelihood directly [34]. The free-energy function is defined as

$$\mathcal{F} = \sum_{n=1}^{N} \sum_{\mathbf{s}_n \in \mathcal{B}} \left(q_n(\mathbf{s}_n; \mathbf{y}_n, \mathbf{W}, \sigma) \log p(\mathbf{y}_n, \mathbf{s}_n, \mathbf{W}, \sigma) + \mathcal{H}[q_n] \right) , \tag{14}$$

where $\mathcal{H}[\cdot]$ refers to the Shannon entropy and $\mathcal{B} = \{0,1\}^H$. If the function $q_n(\cdot)$ is equal to posterior distribution $p(\mathbf{s}_n | \mathbf{y}_n, \mathbf{W}, \sigma)$ then the maxima of the free-energy function correspond to the maxima of the likelihood function. In practical applications this is computationally unfeasible as the calculation of probability distribution over each possible vector $\mathbf{s}_n \in \{0,1\}^H$ for $n = 1, \ldots, K$ is required and each of these values needs to be divided by denominator that is also exponentially hard to compute.

Approximations using the free-energy with some choice for $q_n(\mathbf{s}_n)$ are a popular choice to train latent variable models [34]. In this work we use truncated distributions $q_n(\mathbf{s}_n)$ by applying expectation-truncation scheme (ET) [12]. In this approach, the aim is to consider only the combinations over the most likely values of vector $\mathbf{s}_n$. Truncated approximations are both efficient and precise if most posterior probability mass is, indeed, carried by few values of activations $\mathbf{s}_n$ [14]. For this study, we can expect truncated approximations to be well suited, as we can assume sources with binary values that are sparsely active. We will use approximations with at most γ non-zero entries of $\mathbf{s}_n$. Furthermore, we can further reduce the number of evaluated states by considering only the combinations of the most likely atoms. For many practical applications, direct calculation of a likelihood that a given atom is active may be computationally unfeasible as

$$p(S_{hn} = 1|\mathbf{y}_n) = \sum_{\mathbf{s}_n \in \{\mathbf{s}_n : S_{hn}=1\}} p(\mathbf{s}_n|\mathbf{y}_n) \tag{15}$$

requires a summation over 2^{H-1} binary vectors (one element of $\mathbf{s}_n$ is fixed). Instead of calculating marginals in Equation (15), a selection function is used. The selection function can be any function which has two arguments: observation $\mathbf{y}_n$ and atom $\mathbf{w}_h$. Additionally this function should be efficiently computable and it should provide high values for atoms that were actually used to generate observation $\mathbf{y}_n$. First, H' atoms (candidates) for which the selection function has the highest values for a given observation $\mathbf{y}_n$ are selected (and their indexes h are stored in set $\mathcal{C}_n$) and then $\sum_{\gamma'=0}^{\gamma} \binom{H'}{\gamma'}$ terms with respect to $\mathbf{s}_n$ in Equation (14) are evaluated. Thus, the set of latent states that are taken into account is described as

$$\mathcal{S}_n = \left\{ \mathbf{s} \in \{0,1\}^H : \left(\sum_{h=1}^{H} s_h \leq \gamma \right) \wedge (\forall h \notin \mathcal{C}_n : s_h = 0) \right\} . \tag{16}$$

If it can be assumed that there are at most γ latent variables activated, then for $\mathbf{s}_n \in \mathcal{S}_n$ most of the probability mass of the posterior distribution is contained.

In the case of the BNMF, the selection function can be calculated with formula

$$f(\mathbf{y}_n, \mathbf{w}_h) = -\|\mathbf{y}_n - \max\{\mathbf{w}_h, \mathbf{y}_n\}\|_1 , \tag{17}$$

where $\mathbf{w}_h$ is the h'th column of $\mathbf{W}$ and $\max\{\cdot,\cdot\}$ is an element-wise operation. It was shown in [12] that the selection function in Equation (17) is a lower bound of likelihood ($p(S_{hn} = 1|\mathbf{y}_n)$).

As set $\mathcal{S}_n$ contains only vectors with no more than γ non-zero elements, approximation of the posterior distribution $p(\mathbf{s}_n|\mathbf{y}_n, \mathbf{W}, \sigma^2)$ may be suitable for a subset of observations which were actually generated with no greater than γ number atoms. If an observation is actually generated using more than γ atoms, then approximation of the posterior probability function $p(\mathbf{s}_n|\mathbf{y}_n, \mathbf{W}, \sigma^2)$ will be equal to zero for vectors $\mathbf{s}_n$ with the correct number of active atoms. This may lead to worse estimation of the dictionary. In order to prevent this, only the observations that were actually generated using no more than γ atoms should be taken into account. In [12] it was shown that if we assume that the parameters of the model are already close to optimal parameters, $\sum_{\mathbf{s}_n \in \mathcal{S}_n} p(\mathbf{s}_n, \mathbf{y}|\mathbf{W}, \sigma^2)$ can serve as a function for finding observations that were presumably generated with no more than γ atoms. Thus, set $\mathcal{M}$ can be defined, which contains indexes of N_{cut} observations for which values of sum are largest. Instead of summation over all N observations in Equation (14), only the observations from $\mathcal{M}$ are taken into account.

In the M-step, the truncated posterior distribution is used to obtain an update rule for the parameters. The free energy function is maximized given a fixed truncated posterior (non-zero values of this probability mass function are possible only for $\mathbf{s} \in \mathcal{S}$). The necessary condition is that partial derivatives over all parameters are equal to zero. This condition is satisfied for the dictionary matrix

$$\mathbf{W} = \left(\sum_{n \in \mathcal{M}} \mathbf{y}_n \left\langle \mathbf{s}_n \right\rangle_{\text{ET}}^{\text{T}} \right) \left(\sum_{n \in \mathcal{M}} \left\langle \mathbf{s}_n \mathbf{s}_n^{\text{T}} \right\rangle_{\text{ET}} \right)^{-1} . \tag{18}$$

The expectation $\langle \cdot \rangle_{\text{ET}}$ is over the truncated posterior distribution ($q(\mathbf{s})$). In order to ensure non-negativity of the parameters, matrix $\mathbf{W}$ is updated using a multiplicative update rule

$$\mathbf{W} \leftarrow \mathbf{W} \odot \sum_{n \in \mathcal{M}} \frac{\langle \mathbf{y}_n \mathbf{s}_n^{\text{T}} \rangle_{\text{ET}}}{\mathbf{W} \langle \mathbf{s}_n \mathbf{s}_n^{\text{T}} \rangle_{\text{ET}}} , \tag{19}$$

where $\odot$ is element-wise (Hadamard) multiplication (compare [35], [12]).

IV. Binary non-negative matrix deconvolution

A. Formulation of the model

In this work we propose a new method called binary non-negative matrix deconvolution (BNMD). It is an extension of the binary NMF (see Section III) to a convolutive model. In comparison to the NMD presented in Section II, the proposed model allows only binary activations. Additionally, instead of normal distribution around the approximation (see Equation (12)), the Poisson distribution is used, which is a reasonable choice for audio applications [36], [37]. The objective of the suggested model is to approximate $\mathbf{Y}$ by

$$\hat{\mathbf{Y}} = \sum_{t=1}^{T} \mathbf{W}(t) \overset{t-1\rightarrow}{\mathbf{S}} . \tag{20}$$

The probability density function of the observed spectrogram given activations $\mathbf{S}$ is defined as

$$p(\mathbf{Y}|\mathbf{S}) = \prod_{d=1}^{D} \prod_{n=1}^{N} \mathcal{P}(Y_{dh}; \sum_{h=1}^{H} \sum_{t=1}^{T} \{W_{dh}(t) S_{h,n-t+1}\}) . \tag{21}$$

The prior distribution of the activations in the proposed model is

$$p(\mathbf{S}|\pi) = \prod_{h=1}^{H} \prod_{n=1}^{N} \pi^{S_{hn}} (1-\pi)^{1-S_{hn}} . \tag{22}$$

The free energy function is defined as

$$\mathcal{F} = \sum_{\mathbf{S}\in\mathcal{K}} q(\mathbf{S}) \log p(\mathbf{Y}, \mathbf{S}|\mathbf{W}) + \mathcal{H}[q]\,, \tag{23}$$

where $\mathcal{K}$ is determined similarly to its counterpart in the BNMF (see Equation (16)):

$$\mathcal{K} = \left\{ \mathbf{S} \in \mathcal{E} : \sum_{h=1}^{H}\sum_{n=1}^{N} S_{hn} \leq \gamma \wedge \left(\overset{\forall}{(h,n)} \notin \mathcal{D} : S_{hn} = 0 \right) \right\}\,, \tag{24}$$

where $\mathcal{D}$ is a set of I' pairs (h, n) for which a given selection function gives the lowest values and $\mathcal{E} = \{0, 1\}^{H\times N}$.

B. Segment-wise processing

The number of latent variables can be large for speech processing applications. For example when the number of atoms is 50 and we want to decompose one second of speech spectrogram with frame rate 100 frames per second, then we have $I = H \times N = 50 \cdot 100 = 5000$. Without candidate selection (selection of the atoms that are likely to be active) we would have $\sum_{\gamma'=0}^{15}\binom{5000}{\gamma'} > 10^{43}$ latent states (subset of possible matrices $\mathbf{S}$ with at most $\gamma = 15$ active elements). Even with candidate selection the number of computations would be large. For this reason, a deconvolution is performed in segments. From the spectrogram we extract segments of length L frames sliding with step size S frames. The k'th L-length segment $Y_{d,(k-1)S+l}$ is modeled as

$$\bar{Y}_{d,(k-1)S+l} = \sum_{t=1}^{T}\sum_{i=1}^{I} W_{dh}(t) S_{h,(k-1)S+l-t}, \text{ for } l = 1, \ldots, L \tag{25}$$

We can write it in a matrix form as

$$\mathbf{Y}_k \approx \sum_{t=1}^{T} \mathbf{W}(t) \overset{t-1\rightarrow}{\mathbf{S}_k}\,, \tag{26}$$

where $\mathbf{Y}_k \in \mathbb{R}_{\geq 0}^{D\times L}$ is the k'th segment of the observation matrix and $\mathbf{S}_k \in \{0, 1\}^{H\times L+T-1}$ is the k'th segment of the activation matrix. Note that activation matrix from Equation (20) has the same number of columns as observation matrix $\mathbf{Y}$. The activation matrix for segment $\mathbf{Y}_k$ contains all elements of matrix $\mathbf{S}$ that have influence on elements in segment $\mathbf{Y}_k$. For indexes $k < T$ for non-positive indexes of matrix $\mathbf{S}$ zero values are used.

For the segmented version, the free energy function is

$$\mathcal{F} = \sum_{k\in\mathcal{O}} \sum_{\mathbf{S}_k\in\mathcal{K}_k} q_k(\mathbf{S}_k) \log p(\mathbf{Y}_k, \mathbf{S}_k|\mathbf{W}) + \mathcal{H}[q_k]\,, \tag{27}$$

where $\mathcal{K}_k$ is obtained in the same way as $\mathcal{K}$ but for a given segment k, and $\mathcal{O}$ contains indexes of K_{cut} segments for which values of $\sum_{\mathbf{S}_k\in\mathcal{K}_k} p(\mathbf{S}_k, \mathbf{Y}_k|\mathbf{W}, \sigma^2)$ are largest.

In general, because of the approximations in the E-step, the convergence of the BNMD is not guaranteed. However, experiments have shown that in practice the likelihood does not decrease significantly during the optimization. The detailed description of the experiments can be found in Section VI E.

C. Selection functions

In order to reduce the number of combinations, a selection function is used to select a small subset of the most likely candidates of elements in matrices $\mathbf{S}_k$. Preliminary experiments have shown that the selection function presented in the previous section is not performing sufficiently well. This is probably because some atoms attract all candidates with various shifts. We suggest using cosine similarity as a selection function which is defined as

$$f(\underline{\mathbf{Y}}, \mathbf{W}_h) = \frac{\text{trace}(\underline{\mathbf{Y}}^{\mathrm{T}}\mathbf{W}_h)}{\|\underline{\mathbf{Y}}\|_F \|\mathbf{W}_h\|_F}\,, \tag{28}$$

where

$$\mathbf{W}_h = \begin{bmatrix} W_{1h}(1) & \ldots & W_{1h}(T) \\ \vdots & \ddots & \vdots \\ W_{Dh}(1) & \ldots & W_{Dh}(T) \end{bmatrix} \tag{29}$$

and $\underline{\mathbf{Y}}$ is a T-frame length excerpt of spectrogram $\mathbf{Y}$. In contrast to segments (which can have length different than T), a selection function has to be calculated for all possible excerpts (with 1-frame step). Thus, values of the selection function are calculated for all corresponding elements of matrix $\mathbf{S}$ (for all pairs of indexes $h = 1, \ldots, H$ and $n = 1, \ldots, N$). $\|\cdot\|_F$ denotes the Frobenius norm. This selection function was also tested for dictionary and segment transformed to the log-scale

$$f_{\log} = f(\log \underline{\mathbf{Y}}, \log(\mathbf{W}_h))\,, \tag{30}$$

where logarithms are taken entry-wise.

We have also tested the selection function based on a left inverse of the dictionary matrix ($\mathbf{W}^+$) which has the property that

$$\text{trace}\left(\mathbf{W}_g^+\mathbf{W}_h\right) = \delta_{gh}\,, \tag{31}$$

where δ_{gh} is the Kronecker delta. If $\underline{\mathbf{Y}}$ contains a pattern similar to the h'th atom, a value of trace $\left(\mathbf{W}_{h'}^+\underline{\mathbf{Y}}\right)$ should be close to one if $h = h'$, and close to zero when $h \neq h'$. Matrices $\mathbf{W}_1^+, \ldots, \mathbf{W}_H^+$ were obtained from regularized left inverse matrix

$$\mathbf{W}^+ = (\mathbf{W}^{\mathrm{T}}\mathbf{W} + \xi\mathbf{I})^{-1}\mathbf{W}^{\mathrm{T}} \tag{32}$$

where

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}(1) \\ \vdots \\ \mathbf{W}(T) \end{bmatrix}\,, \tag{33}$$

and ξ is a regularization parameter. In order to obtain $\mathbf{W}_h^+$ from $\mathbf{W}^+$, elements of the h'th row of $\mathbf{W}^+$ are reshaped into $T \times D$ matrix row-wise and transposed. The selection function based on $\mathbf{W}^+$ is calculated using the formula

$$f_{\text{regul}}(\underline{\mathbf{Y}}, \mathbf{W}_h^+) = \frac{\text{trace}(\underline{\mathbf{Y}}^{\mathrm{T}}\mathbf{W}_h^+)}{\|\underline{\mathbf{Y}}\|_F \|\mathbf{W}_h^+\|_F}\,. \tag{34}$$

Selection function $f_{\text{regul}}(\cdot,\cdot)$ (Equation (34)) was also tested with data ($\underline{\mathbf{Y}}$), and dictionary ($\mathbf{W}$ or $\mathbf{W}^+$) represented in the logarithmic scale:

$$f_{\text{log,regul}} = f_{\text{regul}}(\log(\underline{\mathbf{Y}}), (\log(\mathbf{W}_h))^+)\,. \tag{35}$$

D. Parameter estimation

The posterior distribution of $\mathbf{S}_k$ can be obtained using the formula

$$p(\mathbf{S}_k|\mathbf{Y}_k,\mathbf{W}(t)) = \frac{\mathcal{P}(\mathbf{Y}_k;\sum_t \mathbf{W}(t)\overset{t-1\rightarrow}{\mathbf{S}}_k)p(\mathbf{S}_k)}{\sum_{\mathbf{S}'}\mathcal{P}(\mathbf{Y}_k;\sum_t \mathbf{W}(t)\overset{t-1\rightarrow}{\mathbf{S}'_k})p(\mathbf{S}'_k)}\,. \tag{36}$$

The scheme of the method is shown in Figure 1. The analyzed segment spans $L = 15$ frames of the spectrogram. The activation matrix contains $L+T-1 = 15+15-1 = 29$ of possible time instants of atoms that can influence the observed excerpt. During the E-step, the expected value of matrix $\langle\mathbf{S}\rangle_{\mathrm{ET}}$ is computed as

$$\langle\mathbf{S}_k\rangle_{\mathrm{ET}} = \sum_{\mathbf{S}_k\in\mathcal{K}_k} q_k(\mathbf{S}_k)\mathbf{S}_k\,, \tag{37}$$

where

$$q_k(\mathbf{S}_k) = \frac{p(\mathbf{Y}_k|\mathbf{S}_k)p(\mathbf{S}_k)}{\sum_{\mathbf{S}'_k\in\mathcal{K}_k} p(\mathbf{Y}_k|\mathbf{S}'_k)p(\mathbf{S}'_k)}\,. \tag{38}$$

Matrix $\langle\mathbf{S}_k\rangle_{\mathrm{ET}}$ is used to update the dictionary matrix as

$$\mathbf{W}(t) \leftarrow \mathbf{W}(t)\odot\sum_{k\in\mathcal{O}}\frac{\left(\frac{\mathbf{Y}_k}{\hat{\mathbf{Y}}_k}\right)\left[(\overset{t\rightarrow}{\langle\mathbf{S}_k\rangle_{\mathrm{ET}}})\right]^{\mathrm{T}}}{\mathbf{1}\left[\overset{t\rightarrow}{\langle\mathbf{S}_k\rangle_{\mathrm{ET}}})\right]^{\mathrm{T}}}\,. \tag{39}$$

E. Summary of the proposed algorithm

The whole procedure for dictionary learning using binary non-negative matrix deconvolution is as follows:

```
Initialize dictionary ${\mathbf{W}(t)}_{t=1,...,T}$ with random positive values or using VQ atoms (see Section V).
repeat
    for each segment $k \in \{1 \dots K\}$ do
        Compute the selection function for the whole utterance using one of the selection functions (Equation (28), (30), (34), or (35)).
        Compute $I'$ candidates (matrices from set $\mathcal{K}_k$ for which selection function gives lowest values).
        Calculate truncated posteriors for each state in $\mathcal{K}_k$ using Equation (38).
        Compute the expected activation matrix using Equation (37).
        Compute the reconstruction
            $\hat{\mathbf{Y}}_k = \sum_t \mathbf{W}(t)\overset{t-1\rightarrow}{\langle\mathbf{S}_k\rangle}$    (40)
        Store the numerator and denominator from Equation (39) for current $k$.
    end for
    For each segment compute $\sum_{\mathbf{S}_k\in\mathcal{K}_k} p(\mathbf{S}_k,\mathbf{Y}_k|\mathbf{W},\sigma^2)$ and assign to $\mathcal{O}$ a set of $K_{\text{cut}}$ indexes of segments that give highest values.
    Update the dictionary using Equation (39).
until convergence
```

F. Analysis of activations and dictionaries

The dictionary obtained after 20 iterations from the training data for a randomly chosen speaker is shown in Figure 3. In this dictionary most of the atoms represent phoneme-like spectro-temporal patterns as opposed to NMD atoms. For comparison a dictionary learned using the NMD algorithm is shown in Figure 4. This dictionary was estimated by executing the NMD algorithm for 200 iterations. It is visible that there are many atoms with one significant maximum. There are many patterns that are composed of nearly one bin.

One of the visible differences between the dictionaries presented in Figures 3 and 4 is that atoms in the BNMD dictionary are more smooth. When the activation matrix is very sparse the dictionary has to be smooth in order to obtain reconstruction of observation with typical smoothness of speech signals. In the case of not sufficiently sparse activation matrix, it is possible to reconstruct a smooth spectrogram using various temporal shifts of non-smooth atoms. However, setting λ to a large value not necessarily leads to better solution in terms of properties needed in speaker recognition/speech separation. In Figure 4 the presented dictionary was obtained for λ which gave the best speaker identification results. For this λ activation matrix is less sparse than for BNMD, and the atoms are less smooth.

In order to quantify properties of the dictionaries the following concentration measures for each atom were computed:

- Shannon entropy

$$H_h = -\sum_{d=1}^{D}\sum_{t=1}^{T} W_{dh}(t)\log W_{dh}(t)\ , \tag{41}$$

- l2/l1 norm

$$l_h = \frac{\sqrt{\sum_{d=1}^{D}\sum_{t=1}^{T} W_{dh}^2(t)}}{\sum_{d=1}^{D}\sum_{t=1}^{T} W_{dh}(t)}\ , \tag{42}$$

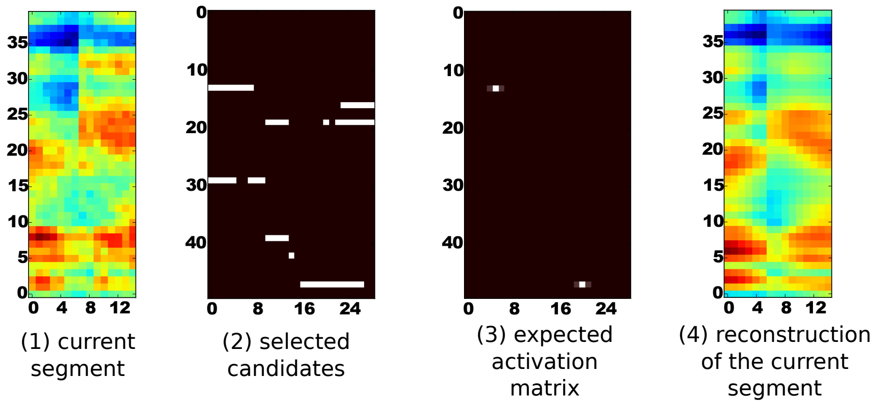
- Gini index

$$g_h = 1 - 2\sum_{i=1}^{DT}\frac{c_i}{\|c_i\|_1}\left(\frac{DT-k+\frac{1}{2}}{DT}\right)\ , \tag{43}$$

where $c_1 \le c_2 \le \ldots \le c_{DT}$ are sorted elements of $W_{dh}(t)$ for $d = 1,\ldots,D$ and $t = 1,\ldots,T$.

The above statistics were averaged over atoms for the BNMD-based dictionary with 50 atoms and NMD dictionary with 50 atoms. They were compared to a dictionary that contains 1000 randomly chosen exemplars from the training dataset. The statistics are plotted in panels of Figure 2. A difference between the BNMD atoms and exemplars is much smaller than the NMD and the exemplars. Thus, concentration characteristics of atoms obtained using the BNMD is close to the characteristics of exemplars which contains realizations of phonemes.

The NMD in comparison to the BNMD can be interpreted as maximum a-posteriori approximation. Moreover, sparsity of the activation matrix is obtained by using an $l1$ regularization term. Therefore, in NMD, the activation matrix can be optimized using effective multiplicative update formula which is related to gradient-based optimization method. In the BNMD,



(1) current segment

(2) selected candidates

(3) expected activation matrix

(4) reconstruction of the current segment

Fig. 1. Approximation of an example spectrogram segment. (1) spectrogram excerpt for the current segment $\mathbf{Y}_k$, (2) selected candidates (elements of set $\mathcal{D}$) are marked with white color, (3) expected activation matrix calculated using Equation (37), (4) reconstruction $\hat{\mathbf{Y}}_k$

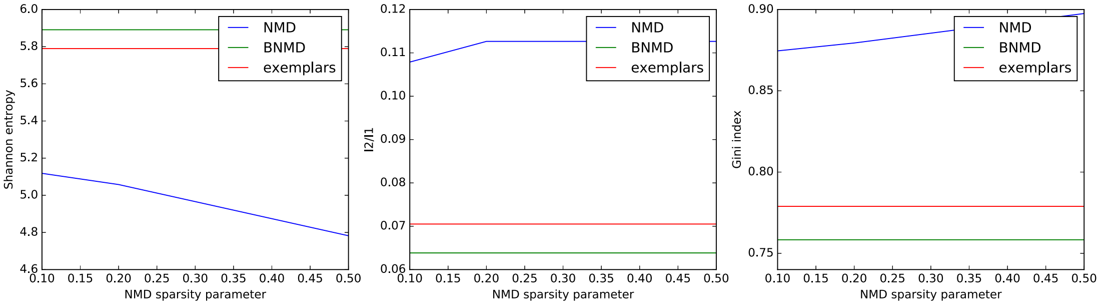


Fig. 2. Comparison of Shannon entropy, l2/l1 and Gini index for BNMD and exemplar-based, and NMD dictionaries. The concentration measures for NMD were obtained for different values of sparsity parameter λ.

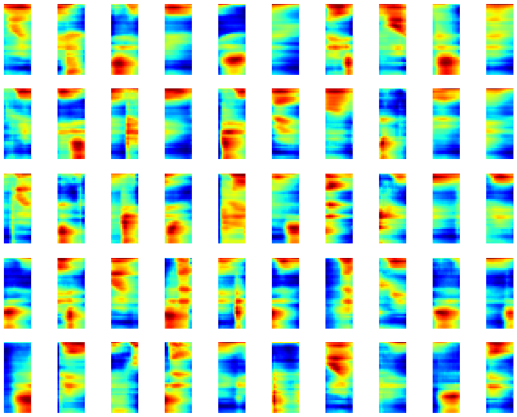


Fig. 3. Dictionary obtained using BNMD method

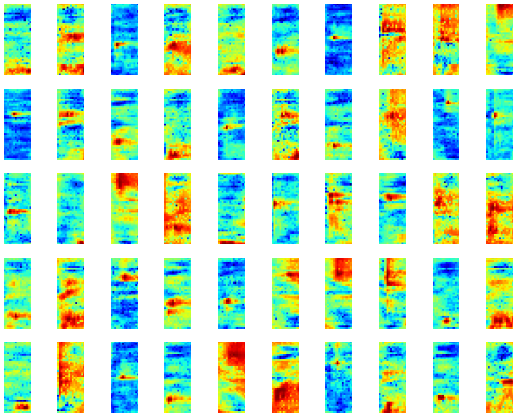


Fig. 4. Dictionary obtained using the NMD method

instead of maximum a-posteriori approximation, expectation-truncation is used, which provides a better approximation of posterior distribution of the activation matrix than point estimate. However, it requires the computation of posterior probabilities for a substantial number of possible activation matrices. ET also a allows for directly the sparsity in terms of the number of active elements.

An analysis was performed to study the typical number of active atoms in a segment when the BNMD was used to represent speech data. We analyzed ET-expected activation matrices ($\langle \mathbf{S}_k \rangle_{\mathrm{ET}}$) from the final training iteration for a randomly chosen speaker in the test data described in Section VI-A. For each segment k the number of elements in $\langle \mathbf{S}_k \rangle_{\mathrm{ET}}$ greater than Θ was counted. In Figure 5, histograms for

$\Theta = k \max(\langle \mathbf{S}_k \rangle_{\text{ET}})$, with $m = 0.3, 0.4, 0.5$ are presented with blue, green and red bars respectively. $\max(\cdot)$ in this case denotes the biggest value in matrix in its argument. It is noticeable that in all cases the number of segments with two significant active elements dominate. It means that for most of the segments, the most probable activation matrices have more than one active elements.

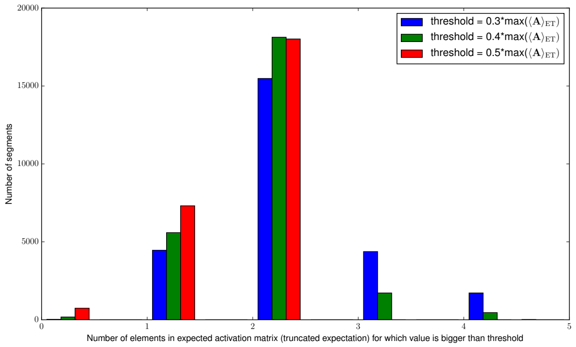


Fig. 5. Number of ET-expected activations above certain threshold.

V. Application of BNMD to speaker recognition and separation

We demonstrate the effectiveness of the proposed binary non-negative matrix deconvolution in a speaker recognition task. In order to apply BNMD dictionaries in speaker recognition, dictionary $\mathbf{W}_s$ is learned for each speaker $s \in 1, \ldots, M$, where M is the number of the speakers. All these dictionaries are adapted starting from a dictionary of a randomly selected speaker. The dictionary for this selected speaker is learned by starting from a VQ dictionary. The VQ dictionary is obtained by applying of k-means algorithm (the version based on Euclidean distance [38] or the version based on KL-divergence - see Section V.B in [39]) to the training dataset. All training vectors for the k-means are obtained using a window spanning T frames sliding with one-frame step from spectrograms $\mathbf{Y}$ that are used also in factorization. After VQ initialization, BNMD iterations (see the procedure in Section IV-E) are done. In this way, described above we obtain dictionaries $\mathbf{W}_1(t), \ldots, \mathbf{W}_M(t)$. During the test, a dictionary that contains atoms of all speakers and noise is built by concatenating speaker-specific dictionaries as

$$\bar{\mathbf{W}}(t) = [\mathbf{W}_1(t) \;\ldots\; \mathbf{W}_M(t) \; \mathbf{W}_{\text{NOISE}}(t)] \;, \tag{44}$$

where $\mathbf{W}_{\text{NOISE}}$ is the dictionary that comprises spectrogram patches that contain noise sampled from the recording before and after the utterance. In order to be able to do the recognition fast, NMD algorithm described in Section II is used during the test time stage. This solves the following optimization problem

$$\begin{aligned} \min_{\mathbf{S}} \quad & KL(\mathbf{Y} \| \sum_{t=1}^{T} \mathbf{W}(t) \overset{t-1\rightarrow}{\mathbf{S}}) + \|\mathbf{\Lambda S}\|_1 \\ \text{s.t.} \quad & \mathbf{S} \geq \mathbf{0} \;, \end{aligned} \tag{45}$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda_{\text{speech}} \mathbf{N}_1 & \ldots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \ldots & \lambda_{\text{speech}} \mathbf{N}_M & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \lambda_{\text{noise}} \mathbf{N}_{\text{NOISE}} \end{bmatrix} , \tag{46}$$

λ_{speech}, λ_{noise} denote sparsity parameters for speech and noise activations respectively, while $\mathbf{N}_1, \ldots, \mathbf{N}_M, \mathbf{N}_{\text{NOISE}}$ are diagonal matrices where h'th diagonal element contains l_1 norm of the h'th atom.

After obtaining the activation matrix using NMD the elements of the activation matrix corresponding to different speakers are summed up. The recognized speaker is the speaker with the highest activation. This method is referred to as maximal activation. The activation mapping procedures proposed in [40] which were demonstrated to enhance speaker recognition accuracy, were also tested. These techniques map activation matrices to matrices that are built using speaker label information, which are called target matrices. For each utterance, a target matrix has M rows (the number of the speakers) and N columns (the number of spectrogram frames). Only the row corresponding to speaker in a given utterance has non-zero values. This row contains RMS-values of signal in utterance's frames. During the training, NMD is used to learn a mapping from activation matrices to target matrices. At the recognition stage, the learned mapping is applied to the activations of a given utterance. A matrix that is a result of the mapping is used to make a decision about the identity of the speaker in the utterance. In [40], the values in the rows were summed up to obtain scores for the considered speakers. In order to obtain a more robust system, square root compression in this matrix can be used prior to summation. We will refer to the former scheme as "mapping (sum.)" and to the latter as "mapping (compr.)".

The BNMD can also be applied to speech separation problem. The dictionary learned using the BNMD was applied in NMD-based speaker-dependent separation task. After optimization of the activation matrix, the spectrogram was reconstructed using the "Wiener-like" filter that is designed using the speech and the noise atoms as in [41]. This filter is a ratio of the spectrogram of the noisy utterance, modeled using activated speech atoms only, to the spectrogram modeled using all atoms.

VI. Experiments

A. Speech database

Experiments have been performed using recordings from the corpus from the 2nd CHiME speech separation and recognition challenge [42]. We used track 1 data from this corpus which contains utterances from 34 speakers. For each speaker, 500 training utterances with the same grammar are available. The training data contains both clean and noisy speech, with SNRs from 9 dB to -6 dB with 3 dB step. The test dataset contains 600 utterances from the same speakers with the same SNRs as in the training dataset. Utterance lengths range from approximately 1.2 to 2.7 s with a mean length of about 1.9 s. The sampling rate used in the corpus is 16000 Hz.

B. Feature extraction

All of the tested algorithms model mel-spectrograms. First, the preemphasis filter with transfer function $H(z) = 1 - 0.97z^{-1}$ was applied. Next, the signal was divided into 25 ms frames and the Hamming window was applied. The frame step was 10 ms. After FFT computation a mel-frequency filterbank was used by multiplication of absolute values of the FFT the triangular filters' characteristics. The 40 filters spanned the frequency range 64–8000 Hz. In the case of the I-vectors and the GMM ML reference methods, cepstral coefficients are extracted by performing two additional steps – calculation of logarithms and type II discrete cosine transform.

C. Baseline systems

The results that we obtained were compared with the speaker recognition accuracies reported in [40], where I-vector, GMM, and template-based (TMPL) were evaluated. The first two mentioned systems are classical speaker recognition methods. In the remaining systems, based on the dictionaries , NMD was used during the test. The dictionaries were obtained using various techniques: EUC VQ (k-means), which is Euclidean error based VQ, KL VQ (KL k-means), Kullback-Leibler distance based VQ, NMD, NMD ES (NMD with early stopping), and the BNMD. The TMPL can be considered as a dictionary learning method. One of the baseline systems was I-vector system used for NIST 2012 [43]. The authors of [43] trained UBM (universal background model) using NIST corpora published before 2012 (NIST SRE 2006, SRE 2008, and SRE 2012). The speech from the CHiME corpus was not applied to train the UBM. PLDA (probabilistic linear discriminant analysis) was also used. Another reference system was a GMM trained for each speaker with maximum likelihood criterion [44]. In the reported experiments, 20 cepstral coefficients were used. The number of the GMM components was 64.

In the template-based systems, the CHiME annotations were used to generate 250 speech atoms per speaker. This dictionary was built using a method described in [45]. In the original work, this kind of dictionary is referred to as exemplar-based. Exemplars are typically obtained by sampling training data, thus in the article this dictionary is called template-based.

In order to obtain the template-based dictionary, transcriptions of speech are needed. They are used to obtain CHiME HTK models, and forced alignment information. Each word in the CHiME corpus is modeled by left-to-right HMM (hidden Markov model). There are two HMM states per phoneme which results in 250 states.

To build template-based dictionary all mel-scale spectrograms from the training dataset were used. For each speaker, for each occurrence of a given state, an excerpt (with size $B \times T$) of the spectrogram was stored. The excerpt has to be temporally centered in a time range spanned by the state label. An element-wise median of all excerpts for a given state form an atom in the template-based dictionary. The BNMD was compared to the system from [45] where the dictionary comprised 250 atoms per speaker and 250 noise atoms, which resulted in 8750 atoms. Additionally, in contrast to the BNMD dictionary, T in template-based dictionary was 25 instead of 15.

During the test time, for each test utterance, an activation matrix is obtained using NMD algorithm. Sparsity parameter λ_{speech} was set to 0.1, while $\lambda_{\text{noise}} = 0.85\lambda_{\text{speech}}$. The next step is to compute a score for each speaker. This was done as described in Section V.

Besides the reference systems from [40], speaker recognition with dictionaries learned by NMD and VQ were performed. The NMD dictionary was obtained using the standard update formulas [20]. At the beginning of each iteration, the columns of $\mathbf{W}(t)$ were normalized to unity l^2-norm. The speaker recognition system was also tested with dictionary obtained using NMD stopped after twenty iterations (system labeled as NMD ES). Two types of VQ algorithms were used as reference dictionary learning: k-means [38] which uses the Euclidean distance error (labeled as EUC VQ). The second type of VQ was KL k-means (see Section V.B in [39]) which uses the Kullback-Leibler divergence as an error measure (labeled as KL VQ). A sliding window with length T was applied to all training utterances. After vectorization this resulted with a set of vectors, which were clustered by means of the mentioned VQ algorithms. The NMD dictionary or dictionaries obtained using VQ (each centroid was an atom) were used for speaker recognition as described in Section V.

D. Parameters of the BNMD system

The BNMD was tested using atoms spanning $T = 15$ frames. This corresponds to duration about 150 ms, which is substantially longer than in typical speaker recognition system based on GMMs and MFCCs. Additionally this duration is sufficient to span most of the phonemes. The segment length was $L = 15$. As atom length is $T = 15$ for $L = 15$ in most of the segments there will be at most two atoms activated. This makes reasonable to set $\gamma = 2$ (highest number of active elements in a segment). Increasing of γ substantially changes the number of elements in $\mathcal{K}_k$ and computational cost. The number of atoms that was tested was $H = 50, 100, \text{and } 250$. Thus it was possible to compare the BNMD with template-based system but also testing more compact dictionaries. The shift of the segment was $S = 1$. The BNMD algorithm was initialized with the dictionary obtained using k-means. The number K_{cut} of the selected segments (whose indexes were in a set assigned to $\mathcal{O}$) was set to $\lfloor 0.8 \cdot K \rfloor$. It was chosen according to the results of preliminary experiments on the same data as the experiments presented in further sections. For this value it is likely that for selected frames the desired latent state is in the $\mathcal{K}$. The parameter I' was 50 and it was set according to the experiments described in Section VI-I. It can be read from Table VI that for $I' = 50$ the in 85% of segments all truly activated atoms are in the set of candidates $\mathcal{D}$.

E. Convergence

In order to practically test the convergence of the BNMD, changes of the log-likelihood during optimization has been analyzed. The tests have been performed for dictionaries with 50 atoms. For each speaker, 100 iterations have been tested.

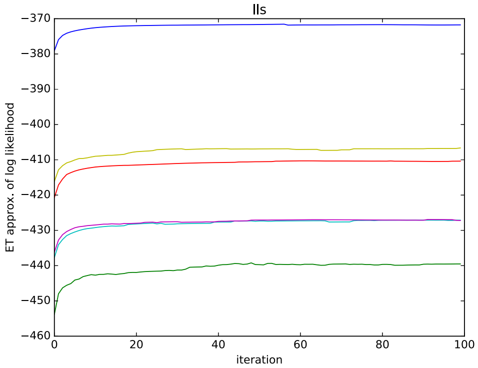


Fig. 6. ET approximated log-likelihood during BNMD iterations.

The log-likelihood did not decrease by more than 0.1% Thus, it can be expected that there will not be a significant decrease. Moreover, decrements occurred occasionally. Thus, we conclude that the BNMD is generally robust in practice, despite the absence of any theoretical guarantees. A graph of truncated log-likelihood during dictionary learning for randomly chosen 6 speakers are shown in Figure 6.

F. Speaker identification results

The results for the speaker identification task obtained using reference methods and the BNMD with dictionary containing 50 atoms are shown in Table I. In the dictionary-based systems the speaker identification is based on maximal activation (see Section V).

I-vector and GMM system performed poorly for lower SNRs. For GMM, the accuracy is 96.7% for 9 dB SNR, and 60.8% for -6 dB, which gives about 37% of relative accuracy decrease. For I-vectors, the relative decrease between the highest and the lowest tested SNRs is 44.8%.

It should be emphasized that the i-vector-based system was obtained using another database. This could lead to a lower accuracy of speaker recognition. Additionally, low accuracy of the i-vector system can be attributed to issues connected with the length of the utterances. It was shown in [46], [47] that for short utterances the variation of i-vectors' elements is higher and the performance of i-vector speaker verification systems degrade rapidly. In the mentioned work several techniques were suggested to improve i-vectors when only short utterances are available. There is also an issue in i-vector based systems, connected the the mismatch in length of utterances in development and test datasets. As the process of finding activations during test, transforming to the representation which is used to make decision, are independent it is on favor of the BNMD.

It can be noticed that NMD-based dictionary brings the lowest accuracy among the tested systems. However, when the dictionary learning is stopped after 20 iterations (NMD ES), then significant improvement can be obtained (average accuracy changes from 37.8% to 75.4%).

TABLE I
ACCURACY OF SPEAKER IDENTIFICATION (%) - DICTIONARY SIZE 50 ATOMS

	9dB	6dB	3dB	0dB	-3dB	-6dB	avg.
I-vect.	62.5	57.0	50.7	46.7	40.3	34.5	48.6
GMM	96.7	91.2	88.7	76.2	68.2	60.8	80.3
EUC VQ	96.8	95.7	93.2	90.0	83.3	68.0	87.8
KL VQ	96.7	95.0	92.8	88.7	81.2	67.3	86.9
NMD	54.3	52.8	46.7	35.2	25.3	12.2	37.8
NMD ES	91.8	88.7	87.3	78.5	63.7	42.5	75.4
BNMD	**98.8**	**98.3**	**97.7**	**93.5**	**88.0**	**71.3**	**91.3**

TABLE II
ACCURACY OF SPEAKER IDENTIFICATION (%) - HIGHER NUMBER OF ATOMS

#atoms	method	9dB	6dB	3dB	0dB	-3dB	-6dB	avg
100	VQ	98.0	97.8	96.0	92.8	84.3	70.0	89.8
	BNMD	**99.5**	**98.8**	**98.0**	**95.5**	**90.7**	**73.8**	**92.6**
250	TMPL	99.0	98.2	97.5	93.2	88.5	69.0	90.9
	VQ	99.1	98.8	98.1	95.0	87.3	68.7	91.2
	BNMD	**99.5**	**99.0**	**98.5**	**95.7**	**89.8**	**72.7**	**92.5**

The highest accuracies were obtained by the BNMD algorithm. The improvement in comparison to VQ is in range from 2% for 9 dB to 4.7% for SNR -3 dB. The relative accuracy decrease between the highest and the lowest tested SNRs is about 30% while for the BNMD it is 28%. It should be emphasized that the BNMD performs better than the template-based system (91.3% in comparison to 90.9%), but it has shorter atoms ($T = 15$ instead of $T = 25$), smaller dictionary ($H = 50$ in comparison to $H = 250$), and annotations are not needed to prepare the dictionary.

Table II summarizes the results for dictionaries with 100 and 250 atoms. For 100 atoms, two methods were compared: VQ and the BNMD. When the number of atoms was changed from 50 to 100, the average accuracy of speaker identification increased from 87.8 to 89.8, while for the BNMD an improvement was achieved from 91.3 to 92.6. For 250-atom dictionary results for VQ the average accuracy is highest than for both smaller dictionaries and it is equal to 91.2. For the BNMD, a slight improvement can be observed to 92.5. The biggest difference between VQ and the BNMD is for SNR -6dB and it is 4%. In the case of dictionary size equal to 250 atoms, the results obtained using template-based dictionary are also presented. The average accuracy is worse than either VQ and the BNMD.

In Table III the results for different mapping types are summarized. It can be noticed that the accuracy increases with the dictionary size. Additionally the difference between the BNMD and VQ is getting smaller, when the number of atoms is increased. In the case of mapping of activation matrix (mapping (sum)) the difference ranges from 2.1% to 0.1%, while for mapping (compr.) it is from 1.7 to 0.4%. The mapping gives substantial improvement over maximal activation (the highest was obtained for template-based system – from 90.9 to 98.6%). The BNMD with mapping can outperform the template-based accuracy, although, as mentioned earlier, the phonetic labels were used in TMPL to build the dictionary and there is a difference in the length of these dictionaries. For the template-based technique atoms have length 25 frames while for the BNMD it is 15.

TABLE III
ACCURACY OF SPEAKER IDENTIFICATION (%) WITH DICTIONARY-BASED METHODS ENHANCED BY ACTIVATIONS MAPPING

n atoms	mapping type	VQ	BNMD	TMPL
50	maximal activation	87.8	**91.3**	–
100	maximal activation	89.8	**92.6**	–
250	maximal activation	91.2	**92.5**	90.9
50	mapping (sum)	95.0	**97.1**	–
100	mapping (sum)	96.7	**98.0**	–
250	mapping (sum)	**98.4**	98.3	98.2
50	mapping (compr.)	95.9	**97.6**	–
100	mapping (compr.)	97.1	**98.4**	–
250	mapping (compr.)	98.3	**98.7**	98.6

TABLE IV
RESULTS FOR SPEECH SEPARATION TASK, SDR (DB)

#atoms		9dB	6dB	3dB	0dB	-3dB	-6dB	avg.
50	VQ	12.2	10.7	9.1	7.4	**5.7**	**4.5**	8.3
	BNMD	**12.7**	**11.0**	**9.3**	**7.6**	**5.7**	4.3	**8.5**
100	VQ	12.5	10.9	9.2	7.4	**5.6**	**4.2**	8.3
	BNMD	**12.9**	**11.1**	**9.4**	**7.5**	5.5	4.0	**8.4**
250	VQ	12.8	**11.5**	9.2	**7.3**	**5.3**	**3.7**	**8.2**
	BNMD	**13.0**	11.2	**9.3**	**7.3**	5.1	3.4	**8.2**

G. Approximated inference

For each utterance, the BNMD algorithm was used to compute approximated (ET) log-likelihood. This was done for each hypothesized speaker. Thus, the number of tests for each trial (test utterance) is equal to the number of speakers in the corpus. In each test the approximated log-likelihood is computed given a dictionary composed of 50 atoms of the hypothesized speaker and 250 noise atoms. The procedure was as follows. For each segment, H' candidates were selected from speech atoms and the second H' from noise atoms. For each segment $\mathbf{Y}$ its model is

$$\hat{\mathbf{Y}} = \sum_{t=1}^{T} \left(\mathbf{W}_{\text{SPEAKER}}(t) \overset{t-1\rightarrow}{\mathbf{S}_{\text{SPEAKER}}} + \mathbf{W}_{\text{NOISE}}(t) \overset{t-1\rightarrow}{\mathbf{S}_{\text{NOISE}}} \right) . \tag{47}$$

Thus, in ET log-likelihood computation, all pairs of $\sum_{i=0}^{\gamma} \binom{H'}{i}$ of $\mathbf{S}_{\text{SPEAKER}}$ and $\sum_{i=0}^{\gamma} \binom{H'}{i}$ of $\mathbf{S}_{\text{NOISE}}$ were taken into account.

The experiments were performed on a reduced dataset. Only the utterances with SNR 0dB were used. There were 100 trials (2-5 trials per speaker). Finally the accuracy obtained using BNMD was 88.11% while for NMD it was 87.12%.

H. Results for speech separation task

The results for the separation task are presented in Table IV. The performance of separation was measured by signal to distortion ratio (SDR) implemented in BSS Eval toolbox [48]. It is visible that for high SNRs (above 6 dB), SDRs of the separated speech increase with the dictionary size. However, this is not the case for smaller SNRs. The average SDR over all the tested SNRs also decreases with the dictionary size. Table V presents SDRs for varying sparsity parameter λ_{speech} are presented. It can be observed that for smaller values of parameter λ_{speech} SDR increases with the dictionary size, while for bigger values of λ_{speech} we can notice the opposite tendency.

TABLE V
AVERAGE SDR OF SEPARATED SPEECH AS THE FUNCTION OF SPARSITY λ AND DICTIONARY SIZE

	50	100	250
0.001	8.31	8.13	7.68
0.025	8.38	8.24	7.89
0.05	8.43	8.32	8.04
0.1	**8.46**	8.40	8.20
0.2	8.35	8.37	8.28
0.4	7.89	8.02	8.07
0.8	6.93	7.19	7.37

I. Evaluation of selection functions

In order to test and compare various selection functions the transcriptions of CHiME2 corpus were used. The baseline hidden Markov model from the CHiME evaluation has 250 states. First, 55 state labels from this pool was selected manually, to obtain a set of states that correspond spectrogram patches that are aligned to centers of phone realizations rather than transitions between them. Next, the spectrogram excerpts centered on occurrences of states were gathered. This was followed by generation of random activation matrices. Temporal overlap $(0.3 \cdot T)$ of atoms is allowed. More specifically, the element of the first column was selected randomly. The next column was c selected randomly from the range $(T - \lfloor 0.3 \cdot T \rfloor; T)$. The active element for column c was chosen randomly. The process was repeated to fill the activation matrix with 150 columns (the number that is likely to be a number of columns of a spectrogram from an utterance from the CHiME corpus). The generated activation matrix was convolved with randomly chosen realization of states gathered from data. After data preparation, dictionary was computed. Each atom was obtained by averaging all spectrogram excerpts that correspond to a given state label from transcriptions.

With activation matrices and the dictionary it was possible to compare selection functions. Three selection functions were used: cosine similarity, cosine similarity in logarithmic scale, and cosine similarity in logarithmic scale with rows of regularized pseudoinverse in Equation (34).

For each segment 50 elements of the corresponding activation matrix were selected, for whom values of the selection function were highest. Next, it was checked if the elements from the ground truth activation matrix (the activation matrix that was used to synthesize the data) are selected. We checked what was the fraction of segments in which all true activations were selected.

The results are shown in Table VI. The best results were obtained for $f_{\text{log,regul}}$, where the regularization ξ coefficient was set to 64. In general, in logarithmic scale selection function performs better. This gives a compression that prevents time-frequency bins with the highest magnitude to dominate a value of the selection function. Additionally, in the pseudoinverse-based selection functions the information about the whole dictionary is used during detection of a given atom. Regularization of pseudoinverse turned out to be effective in preventing of overfitting.

TABLE VI
COMPARISON OF QUALITY OF SELECTION FUNCTIONS TESTED ON ARTIFICIALLY SYNTHESIZED DATA

selection function	result
cosine similarity (Eq. (28))	55.5
cosine similarity log scale (Eq. (30))	66.6
selection function $f_{\log,\mathrm{regul}}$ (Eq. (35)) $\xi = 32$	84.1
selection function $f_{\log,\mathrm{regul}}$ (Eq. (35)) $\xi = 64$	85.0
selection function $f_{\log,\mathrm{regul}}$ (Eq. (35)) $\xi = 128$	83.8

VII. CONCLUSIONS

This article describes the binary non-negative matrix deconvolution (BNMD), a new method for dictionary learning. The BNMD model corresponds to the sparse non-negative matrix deconvolution, where entries in the activation matrix are constrained to be binary. Additionally, the proposed model uses probabilistic inference to find out which combinations of temporally shifted atoms are likely to explain the analyzed spectrogram.

It was shown that by using the expectation-truncation scheme, it is possible to effectively learn dictionary for this model. Dictionaries learned using the binary non-negative matrix deconvolution were evaluated in NMD-based speaker recognition and speech separation systems. The experiments were performed using 2nd CHiME track 1 data.

The experimental results show that using dictionaries learned with the BNMD give better speaker identification accuracy than the GMM and I-vector systems. This is most notable for lower SNRs, where for GMM accuracy was 68% while for deconvolution with 50-atom BNMD dictionary 88%. The improvement was also observed in other dictionary learning methods namely NMD or vector quantization, which was applied to the spectrograms using a sliding window approach. It has turned out that dictionaries obtained using the vector quantization are better than dictionaries learned by the NMD (accuracies averaged over tested SNRs were 87.8% and 75.4%, respectively). The BNMD dictionary in this test condition gave 91.3%. This improvement over other methods is most significant for the smallest of tested dictionary sizes. As the number of atoms is increased the improvement is getting smaller. The BNMD dictionary also outperforms a dictionary built from templates extracted using phonetic labels. For 250-atom dictionary size BNMD yielded 92.5%, while the template-based dictionary gave 90.9%.

In the case of the speech separation task, dictionaries learned using BNMD give slightly better SDRs on average in comparison to VQ dictionaries (8.5 in comparison to 8.3 dB for 50-atom dictionary). However, higher SDRs can be observed only for higher tested SNRs (-3 – 9 dB). For example the 50-atom BNMD dictionary gave a 0.5 dB higher SDR for 9dB SNR but 0.2 lower SDR for -3 dB SNR. Similarly to the speaker recognition results, the difference between VQ and BNMD dictionaries is getting smaller when the dictionary size is increased.

ACKNOWLEDGMENT

This article was financially supported within the project ”Engineer of the Future. Improving the didactic potential of the Poznan University of Technology” - POKL.04.03.00-00-259/12, implemented within the Human Capital Operational Programme, co-financed by the European Union within the European Social Fund. Furthermore, JL acknowledges funding by the DFG grant EXC 1077/1 (Hearing4all).

REFERENCES

[1] A. Ozerov, E. Vincent, and F. Bimbot, “A general flexible framework for the handling of prior information in audio source separation,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 4, pp. 1118–1133, 2012.

[2] P. Smaragdis, “Convolutive speech bases and their application to supervised speech separation,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 1, pp. 1–12, 2007.

[3] A. Hurmalainen, “Robust speech recognition with spectrogram factorisation,” Ph.D. dissertation, Tampere University of Technology, 2014.

[4] D. Baby, T. Virtanen, J. F. Gemmeke *et al.*, “Coupled dictionaries for exemplar-based speech enhancement and automatic speech recognition,” *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 11, pp. 1788–1799, 2015.

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

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

[7] J. F. Gemmeke, A. Hurmalainen, and T. Virtanen, “HMM-regularization for NMF-based noise robust ASR,” *Proc. CHiME-2013, Vancouver, Canada*, pp. 47–52, 2013.

[8] E. Benetos and S. Dixon, “Multiple-instrument polyphonic music transcription using a temporally constrained shift-invariant model,” *The Journal of the Acoustical Society of America*, vol. 133, no. 3, pp. 1727–1741, 2013.

[9] T. Heittola, A. Klapuri, and T. Virtanen, “Musical instrument recognition in polyphonic audio using source-filter model for sound separation.” in *ISMIR*, 2009, pp. 327–332.

[10] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, “Exemplar-based sparse representations for noise robust automatic speech recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 7, pp. 2067–2080, 2011.

[11] A. Diment and T. Virtanen, “Archetypal analysis for audio dictionary learning,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2015.

[12] J. Lücke and J. Eggert, “Expectation truncation and the benefits of preselection in training generative models,” *The Journal of Machine Learning Research*, vol. 11, pp. 2855–2900, 2010.

[13] M. Henniges, G. Puertas, J. Bornschein, J. Eggert, and J. Lücke, “Binary sparse coding,” in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2010, pp. 450–457.

[14] A. S. Sheikh, J. A. Shelton, and J. Lücke, “A truncated EM approach for spike-and-slab sparse coding,” *JMLR*, vol. 15, pp. 2653–2687, 2014.

[15] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

[16] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” *IEEE Transactions on information theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[17] A. J. Bell and T. J. Sejnowski, “The independent components of natural scenes are edge filters,” *Vision research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[18] A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural networks*, vol. 13, no. 4, pp. 411–430, 2000.

[19] B. A. Olshausen and D. J. Field, “Sparse coding with an overcomplete basis set: A strategy employed by v1?” *Vision research*, vol. 37, no. 23, pp. 3311–3325, 1997.

[20] T. Virtanen, J. F. Gemmeke, B. Raj, and P. Smaragdis, “Compositional models for audio processing,” *IEEE Signal Processing Magazine*, 2015.

[21] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in neural information processing systems*, 2001, pp. 556–562.

[22] M. N. Schmidt and R. K. Olsson., “Single-channel speech separation using sparse non-negative matrix factorization,” in *Proceedings of the 9th International Conference on Spoken Language Processing*, 2006.

[23] A. T. Cemgil, “Bayesian inference for non-negative matrix factorisation models,” *Computational Intelligence and Neuroscience*, 2009.

[24] P. O. Hoyer, “Non-negative sparse coding,” in *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*. IEEE, 2002, pp. 557–565.

[25] I. J. Goodfellow, A. Courville, and Y. Bengio, “Scaling up spike-and-slab models for unsupervised feature learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1902–1914, 2013.

[26] J. Gemmeke, L. ten Bosch, L. Boves, and B. Cranen, “Using sparse representations for exemplar based continuous digit recognition,” in *Signal Processing Conference, 2009 17th European*. IEEE, 2009, pp. 1755–1759.

[27] P. Smaragdis, “Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs,” in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2004, pp. 494–499.

[28] T. Virtanen, “Separation of sound sources by convolutive sparse coding,” in *ISCA Tutorial and Research Workshop (ITRW) on Statistical and Perceptual Audio Processing*, 2004.

[29] P. D. O’Grady and B. A. Pearlmutter., “Discovering convolutive speech phones using sparseness and non-negativity constraints,” in *Proceedings of the 7th International Conference on Independent Component Analysis and Signal Separation (ICA)*, 2007.

[30] S. Kirbiz, A. T. Cemgil, and B. Günsel, “Bayesian inference for non-negative matrix factor deconvolution models,” in *Pattern Recognition (ICPR), 2010 20th International Conference on*. IEEE, 2010, pp. 2812–2815.

[31] O. Dikmen and C. Févotte, “Maximum marginal likelihood estimation for non-negative dictionary learning in the gamma-poisson model,” *Signal Processing, IEEE Transactions on*, vol. 60, no. 10, pp. 5163–5175, 2012.

[32] M. Haft, R. Hofman, and V. Tresp, “Generative binary codes,” *Pattern Anal Appl*, pp. 269–84.

[33] J. Bornschein, M. Henniges, and J. Lücke, “Are v1 simple cells optimized for visual occlusions? a comparative study,” *PLoS Comput Biol*, vol. 9, no. 6, p. e1003062, 2013.

[34] R. Neal and G. Hinton, “A view of the EM algorithm that justifies incremental, sparse, and other variants,” in *Learning in Graphical Models*, M. I. Jordan, Ed. Kluwer, 1998.

[35] D. D. Lee and H. S. Seung, “Learning the parts of objects by non-negative matrix factorization,” *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[36] 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.

[37] B. King, C. Févotte, and P. Smaragdis, “Optimal cost function and magnitude power for NMF-based speech separation and music interpolation,” in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*. IEEE, 2012, pp. 1–6.

[38] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA., 1967, pp. 281–297.

[39] T. Virtanen, J. F. Gemmeke, and B. Raj, “Active-set newton algorithm for overcomplete non-negative representations of audio,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 21, no. 11, pp. 2277–2289, 2013.

[40] A. Hurmalainen, R. Saeidi, and T. Virtanen, “Noise robust speaker recognition with convolutive sparse coding,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[41] B. Raj, T. Virtanen, S. Chaudhuri, and R. Singh, “Non-negative matrix factorization based compensation of music for automatic speech recognition.” in *Interspeech*. Citeseer, 2010, pp. 717–720.

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

[43] R. Saeidi, K.-A. Lee, T. Kinnunen, T. Hasan, B. Fauve, P.-M. Bousquet, E. Khoury, P. Sordo Martinez, J. M. K. Kua, C. You *et al.*, “I4U submission to NIST SRE 2012: A large-scale collaborative effort for noise-robust speaker verification,” 2013.

[44] D. A. Reynolds, “Speaker identification and verification using gaussian mixture speaker models,” *Speech communication*, vol. 17, no. 1, pp. 91–108, 1995.

[45] R. Saeidi, A. Hurmalainen, T. Virtanen, and D. A. van Leeuwen, “Exemplar-based sparse representation and sparse discrimination for noise robust speaker identification.” in *Odyssey*. Citeseer, 2012, pp. 248–255.

[46] A. Kanagasundaram, D. Dean, S. Sridharan, J. Gonzalez-Dominguez, J. Gonzalez-Rodriguez, and D. Ramos, “Improving short utterance i-vector speaker verification using utterance variance modelling and compensation techniques,” *Speech Communication*, vol. 59, pp. 69 – 82, 2014. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0167639314000053

[47] A. Kanagasundaram, D. Dean, J. González Domínguez, S. Sridharan, D. Ramos, and J. Gonzalez-Rodriguez, “Improving short utterance based i-vector speaker recognition using source and utterance-duration normalization techniques,” in *Interspeech*. International Speech Communication Association, 2013.

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