A FACTORIAL SPARSE CODER MODEL FOR SINGLE CHANNEL SOURCE SEPARATION

Robert Peharz [1], *Michael Stark* [1], *Franz Pernkopf* [1], *Yannis Stylianou* [2]

[1]Signal Processing and Speech Communication Lab, Graz University of Technology
[2]Computer Science Department, University of Crete

Abstract

We propose a probabilistic factorial sparse coder model for single channel source separation in the magnitude spectrogram domain. The mixture spectrogram is assumed to be the sum of the sources, which are assumed to be generated frame-wise as the output of sparse coders plus noise. For dictionary training we use an algorithm which can be described as non-negative matrix factorization with ℓ^0 sparseness constraints. In order to infer likely source spectrogram candidates, we approximate the intractable exact inference by maximizing the posterior over a plausible subset of solutions. We compare our system to the factorial-max vector quantization model, where the proposed method shows a superior performance in terms of signal-to-interference ratio. Finally, the low computational requirements of the algorithm allows close to real time applications.

Index Terms: source separation, sparse coding, sparse NMF

1. Introduction

Single channel source separation (SCSS) aims to extract several source signals from a single mixture recording. Since at least two sources are interfering, the SCSS problem is ill-posed and standard source separation methods (e.g. [1]) can not be applied. Although sound sources may overlap in time, they rarely interfere in a time-frequency representation. This fact has been used in computational auditory scene analysis [2, 3], inspired by the human ability to organize the perceived time-frequency representation according to likely sources. Roweis [4] introduced the refiltering framework which uses so-called spectrogram masks in order to attenuate spectrogram parts which do not belong to the desired sources. To estimate these mask signals, he proposed the factorial-max vector quantizer (VQ) model, which assumes that the magnitude-log source spectrograms are generated by vector quantizers plus a noise term. In order to train speaker specific code-books and to estimate the noise variances he applied k-means to source specific spectrograms. Hence, max-VQ explicitly models the sources in a training stage.

In this paper, we extend the factorial-max VQ model by replacing the vector quantizers with sparse coders. A sparse coder can be seen as a generalization of a vector quantizer, since it represents data with a linear combination of up to L so-called atoms (L being a parameter to chose), while a vector quantizer uses a single, non-scalable code-word. Consequently, we call our system factorial sparse coder model (factorial SC). In order to train speaker specific dictionaries, we use a non-negative matrix factorization algorithm with ℓ^0-sparseness constraints on the coefficient matrix (NMFℓ^0).

The paper is organized as follows: In Section 2 we review the factorial-max VQ system. In Section 3 we discuss NMFℓ^0, our training algorithm for non-negative dictionaries. In Section 4 we introduce the factorial sparse coder model and its inference method. Experimental results are discussed in Section 5. Finally, Section 6 concludes the paper.

2. Factorial-Max VQ Model

The factorial-max VQ model [4], here discussed for the case of two interfering sources, usually works in the log magnitude spectral domain. We assume that the spectrogram frames are independent of each other over time, i.e. we work in frame-wise manner. Let $\mathbf{x}$, $\mathbf{s}^1$ and $\mathbf{s}^2$ be spectrogram frames of the mixture and the sources, respectively. The observation model of the speech mixture $\mathbf{x}$ is given as $p(\mathbf{x}|\mathbf{s}^1, \mathbf{s}^2) = \mathcal{N}(\mathbf{x}; \max(\mathbf{s}^1, \mathbf{s}^2), \Sigma)$, where $\mathcal{N}$ is the normal distribution and Σ is a covariance matrix. The mixed signal $\mathbf{x}$ is assumed to be the element-wise maximum of the source log-spectra $\mathbf{s}^1$ and $\mathbf{s}^2$ plus an additive Gaussian noise term. The sources are modeled with vector quantizers with speaker specific code-books $\mathbf{W}^1$ and $\mathbf{W}^2$. The hidden variables z^1 and z^2 select the source spectra from the code-books, i.e. $\mathbf{s}^m = \mathbf{w}^m_{z^m}$, where $\mathbf{w}^m_{z^m}$ is the $z^{m\text{ th}}$ column of $\mathbf{W}^m$, $m \in \{1, 2\}$. According to Bayes theorem, the posterior probability is given as $p(z^1, z^2|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{s}^1, \mathbf{s}^2)\, p(z^1)\, p(z^2)}{p(\mathbf{x})}$. The code-books $\mathbf{W}^1$ and $\mathbf{W}^2$ are trained by applying k-means to speaker specific spectrograms. Additionally, the noise covariance matrix Σ and the prior distributions $p(z^1)$ and $p(z^2)$ are estimated from the output of k-means. In the separation step the combination of indices z^1 and z^2, which maximizes the posterior, is inferred. The corresponding code-words $\mathbf{s}^1 = \mathbf{w}^1_{z^1}$ and $\mathbf{s}^2 = \mathbf{w}^2_{z^2}$ are approximations of the source spectrograms, which are used to calculate the spectrogram masks.

3. Non-negative Matrix Factorization with ℓ^0 Constraints

Aharon et al. [5] proposed the K-SVD algorithm for dictionary training for sparse coders. A sparse coder aims to approximate a D-dimensional vector $\mathbf{x}$ using a linear combination of maximal L so-called signal atoms, which are stored in the columns of a $D \times K$ dictionary matrix $\mathbf{W}$, where usually $L \ll K$. Extending the problem to a set of input vectors arranged in the columns of a data matrix $\mathbf{X}$, we can define the task as minimization of the objective

$$E = \|\mathbf{X} - \mathbf{W}\,\mathbf{H}\|_F^2\,,\ \text{s.t. } L_0(\mathbf{h}_i) \leq L,\ \forall i \qquad (1)$$

with respect to the coefficient matrix $\mathbf{H}$, where $\mathbf{h}_i$ is the i^{th} column of $\mathbf{H}$, $\|\cdot\|_F$ is the Frobenius norm and $L_0(\cdot)$ denotes the ℓ^0-pseudo-norm, i.e. the number of non-zero entries in the argument vector. Unfortunately, the sparse coding problem is NP-hard [6], so that we have to resort to approximate solutions such

as orthogonal matching pursuit (OMP) [7], basis pursuit (BP) [8] or the focal under-determined system solver (FOCUSS) [9].

K-SVD and its nonnegative variant [5] are iterative two stage algorithms which alternate between a sparse coding stage and a dictionary update stage. Similar to K-SVD we proposed a two stage algorithm which we call non-negative matrix factorization with ℓ^0-sparseness constraints (NMFℓ^0) [10]. For the sparse coding stage we proposed non-negative matching pursuit (NMP), a non-negative variant of OMP, which is shown in Algorithm 1. Without loss of generality, we assume that the columns

Algorithm 1 Non-negative Matching Pursuit (NMP)

1: $\mathbf{z} = [\,]$
2: $\mathbf{c} = [\,]$
3: $\mathbf{r} \leftarrow \mathbf{x}$
4: **for** $l = 1 : L$ **do**
5: $\quad \mathbf{a} = \mathbf{W}^{\mathbf{T}}\mathbf{r}$
6: $\quad z^* = \arg\max \mathbf{a}$
7: $\quad c^* = \max \mathbf{a}$
8: $\quad$ **if** $c^* \leq 0$ **then**
9: $\quad\quad$ Terminate
10: $\quad$ **end if**
11: $\quad \mathbf{z} \leftarrow [\mathbf{z}, z^*]$
12: $\quad \mathbf{c} \leftarrow [\mathbf{c}, c^*]$
13: $\quad$ **for** $j = 1 : J$ **do**
14: $\quad\quad \mathbf{c} \leftarrow \mathbf{c} \otimes \frac{(\mathbf{W}_{\mathbf{z}}^T \mathbf{x})}{(\mathbf{W}_{\mathbf{z}}^T \mathbf{W}_{\mathbf{z}}\, \mathbf{c})}$
15: $\quad$ **end for**
16: $\quad \mathbf{r} \leftarrow \mathbf{x} - \mathbf{W}_{\mathbf{z}}\mathbf{c}$
17: **end for**

of $\mathbf{W}$ are normalized to unit length. The algorithm starts with empty index and coefficient vectors $\mathbf{z}$ and $\mathbf{c}$, and assigns the data vector $\mathbf{x}$ to the residual $\mathbf{r}$. In step 6 we select the index z^* of the atom which approximates the residual best, using a most probably positive coefficient c^*. However, for the case that c^* is negative, the algorithm terminates. In steps 13-15 the data $\mathbf{x}$ is approximated using the sub-dictionary $\mathbf{W}_{\mathbf{z}}$ containing the atoms indexed by $\mathbf{z}$. For this task we use the non-negative matrix factorization (NMF) update rule for the coefficient matrix (see [11] and Eq. (2), left). Note that this step delivers non-negative least-squares coefficients $\mathbf{c}$, given that the number of NMF iterations J is sufficiently large. In order to obtain the coefficient matrix $\mathbf{H}$ (with $L_0(\mathbf{h}_i) \leq L, \forall i$), Algorithm 1 has to be repeated for each column in $\mathbf{X}$. The corresponding columns in $\mathbf{H}$ are built by setting the entries depicted by $\mathbf{z}$ to the values stored in $\mathbf{c}$, and setting all other entries to zero.

In the dictionary update step we use several iterations of non-negative matrix factorization (NMF) proposed by Lee and Seung [11]:

$$\mathbf{H} \leftarrow \mathbf{H} \otimes \frac{(\mathbf{W}^T\mathbf{X})}{(\mathbf{W}^T\mathbf{W}\,\mathbf{H})}, \quad \mathbf{W} \leftarrow \mathbf{W} \otimes \frac{(\mathbf{X}\,\mathbf{H}^T)}{(\mathbf{W}\,\mathbf{H}\,\mathbf{H}^T)}. \quad (2)$$

The symbols $\otimes$ and $\frac{\cdot\cdot\cdot}{\cdot\cdot\cdot}$ in Eq. (2) denote element-wise multiplication and division, respectively. Lee and Seung showed that the update rules achieve a local minimum of the objective $\|\mathbf{X} - \mathbf{W}\,\mathbf{H}\|_F^2$, and that non-negativity is maintained. Further, these update rules also have a property which we call *sparseness maintenance*. Since the updates consist of element-wise multiplications, an entry in $\mathbf{W}$ or $\mathbf{H}$, which is zero before an update, is also zero afterwards. Therefore, we simply can use several iterations of the rules in Eq. (2), without destroying the sparse structure of $\mathbf{H}$. NMFℓ^0 is summarized in Algorithm 2,

Algorithm 2 NMFℓ^0

1: Initialize $\mathbf{W}$ randomly
2: **for** $i = 1 : I$ **do**
3: $\quad \mathbf{H} \leftarrow$ sparsely code $\mathbf{X}$ with $\mathbf{W}$ using NMP
4: $\quad$ **for** $j = 1 : J$ **do**
5: $\quad\quad \mathbf{W} \leftarrow \mathbf{W} \otimes \frac{(\mathbf{X}\,\mathbf{H}^T)}{(\mathbf{W}\,\mathbf{H}\,\mathbf{H}^T)}$
6: $\quad\quad \mathbf{w}_k \leftarrow \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|},\ k = 1, \ldots, K$
7: $\quad\quad \mathbf{H} \leftarrow \mathbf{H} \otimes \frac{(\mathbf{W}^T\mathbf{X})}{(\mathbf{W}^T\mathbf{W}\,\mathbf{H})}$
8: $\quad$ **end for**
9: **end for**

where I and J denote the number of overall iterations and NMF updates. The dictionary matrix $\mathbf{W}$ is initialized with randomly selected and normalized data vectors out of $\mathbf{X}$. For the case that an atom is not used at all, i.e. when the corresponding row in $\mathbf{H}$ contains only zeros, we re-initialize this atom using the data vector with largest approximation error.

4. Factorial Sparse Coder Model

The factorial sparse coder (SC) model is a generalization of factorial-max VQ, where we work in the *linear* magnitude spectrogram domain. As in max VQ, we assume that the spectrogram columns are independent of each other over time. The factorial SC model is shown in Figure 1 for the case of two interfering sources, although it can be easily extended to more sources. The mixture spectrogram is assumed to be the sum of

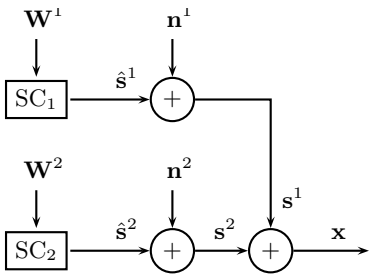

Figure 1: Factorial Sparse Coder Model.

the sources, which are modeled as the output of sparse coders plus noise terms:

$$\mathbf{x} = \sum_{m=1}^{2} \mathbf{s}^m = \sum_{m=1}^{2} (\hat{\mathbf{s}}^m + \mathbf{n}^m). \quad (3)$$

The output of the m^{th} sparse coder is given as

$$\hat{\mathbf{s}}^m = \sum_{k=1}^{L} h^m_{z^m_k} \mathbf{w}^m_{z^m_k} = \mathbf{W}^m_{\mathbf{z}^m}\, \mathbf{h}^m_{\mathbf{z}^m} = \mathbf{W}^m\, \mathbf{h}^m, \quad (4)$$

where $\mathbf{W}^m$ is a source specific dictionary, $\mathbf{h}^m$ is the corresponding coefficient vector and $\mathbf{z}^m$ is an index vector indicating the selected atoms. The right hand side of Eq. (4) holds, since all entries in $\mathbf{h}^m$ not addressed by $\mathbf{z}^m$ are zero. The dictionaries are obtained by applying NMFℓ^0 to source specific spectrogram data. In order to perform source separation, we have to infer coefficient vectors $\mathbf{h}^m$ for the given mixture $\mathbf{x}$, $m \in \{1, 2\}$. To simplify the problem, we assume that the coefficient *values*

are given by a non-negative least-squares approximation of the speech mixture, using the selected atoms of all sources:

$$\begin{pmatrix}\mathbf{h}^1_{\mathbf{z}^1}\\ \mathbf{h}^2_{\mathbf{z}^2}\end{pmatrix} = \arg\min_{\mathbf{h}} \left\|\mathbf{x} - \begin{pmatrix}\mathbf{W}^1_{\mathbf{z}^1} \mathbf{W}^2_{\mathbf{z}^2}\end{pmatrix} \mathbf{h}\right\|^2, \ \forall i : h_i \geq 0. \tag{5}$$

$\begin{pmatrix}\mathbf{W}^1_{\mathbf{z}^1} \mathbf{W}^2_{\mathbf{z}^2}\end{pmatrix}$ is the concatenation of the sub-dictionaries, and on the left hand side of Eq. (5) we have the corresponding stacked coefficients. In this way, the separation algorithm is reduced to a search problem, in order to find suited atoms for each source (i.e. the index vectors $\mathbf{z}^m$, or the locations of the "non-zeros" in $\mathbf{h}^m$). The non-negative least-squares approximation of $\mathbf{x}$ is given as

$$\hat{\mathbf{x}} = \begin{pmatrix}\mathbf{W}^1_{\mathbf{z}^1} \mathbf{W}^2_{\mathbf{z}^2}\end{pmatrix} \begin{pmatrix}\mathbf{h}^1_{\mathbf{z}^1}\\ \mathbf{h}^2_{\mathbf{z}^2}\end{pmatrix} = \sum_{m=1}^{2} \hat{\mathbf{s}}^m, \tag{6}$$

and further, with Eq. (3) we see that

$$\mathbf{x} = \hat{\mathbf{x}} + \sum_{m=1}^{2} \mathbf{n}^m. \tag{7}$$

We assume Laplacian distributed noise in our model, since we observed that the residual error of NMFℓ^0 is distributed according to a Laplace distribution in each frequency bin [10]. The Laplacian form factors $\boldsymbol{\lambda}^m = (\lambda^m_1, \lambda^m_2, \ldots, \lambda^m_D),\ m \in \{1, 2\}$ can be estimated from the residual error in the training stage, where D denotes the number of frequency bins. Since the overall noise is the sum of the individual noise terms in Eq. (7), the probability density function (pdf) of the overall noise is the convolution of the individual pdfs. Therefore, assuming independence among all frequency bins, the d^{th} bin of the mixture is distributed according to a convolution of two Laplacian pdfs with mean value $\hat{x}_d$ (see [10] for a derivation):

$$p(x_d|\hat{x}_d, \lambda^1_d, \lambda^2_d) = \frac{1}{2}\left[\frac{\lambda^1_d\, e^{\frac{-|x_d - \hat{x}_d|}{\lambda^1_d}}}{(\lambda^1_d)^2 - (\lambda^2_d)^2} + \frac{\lambda^2_d\, e^{\frac{-|x_d - \hat{x}_d|}{\lambda^2_d}}}{(\lambda^2_d)^2 - (\lambda^1_d)^2}\right]. \tag{8}$$

Using Eq. (8), the likelihood of $\mathbf{x}$ is given as

$$p(\mathbf{x}|\mathbf{z}^1, \mathbf{z}^2, \boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2) = \prod_{d=1}^{D} p(x_d|\hat{x}_d(\mathbf{z}^1, \mathbf{z}^2), \lambda^1_d, \lambda^2_d), \tag{9}$$

where $\hat{x}_d(\mathbf{z}^1, \mathbf{z}^2)$ indicates that $\hat{\mathbf{x}}$ is a function of $\mathbf{z}^1$ and $\mathbf{z}^2$ (see Eq. (6)). We approximate the prior probability $p(\mathbf{z}^m)$ with a Markov chain according to

$$p(\mathbf{z}^m) = p(z^m_1, \ldots, z^m_L) \approx p(z^m_1) \prod_{k=2}^{L} p(z^m_k|z^m_{k-1}), \tag{10}$$

where the factors $p(z^m_1)$ and $p(z^m_k|z^m_{k-1})$ can be estimated from the coefficient matrix returned by NMFℓ^0. Using Bayes theorem and assuming independent sources, the posterior probability of the index vectors $\mathbf{z}^1$ and $\mathbf{z}^2$ for a given mixture spectrum $\mathbf{x}$ is given as

$$p(\mathbf{z}^1, \mathbf{z}^2|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z}^1, \mathbf{z}^2, \boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2) p(\mathbf{z}^1)\, p(\mathbf{z}^2)}{p(\mathbf{x})}. \tag{11}$$

Hence, source separation is achieved by finding $\mathbf{z}^1$ and $\mathbf{z}^2$ which maximize Eq. (11), where the normalization term $p(\mathbf{x})$ can be neglected in optimization.

However, maximization of Eq. (11) is not easy, and brute-force search would consider $\binom{K}{L}^2$ combinations, assuming that each dictionary comprises K atoms. Therefore, we propose a multi-hypotheses variant of matching pursuit which constricts the considered solutions to a plausible sub-set. Our inference method is described in Algorithm 3. First, we concatenate the source specific dictionaries: $\mathbf{W} := \begin{pmatrix}\mathbf{W}^1 \mathbf{W}^2\end{pmatrix}$. A solution is defined as a triplet $\xi = \langle \mathbf{z}, \mathbf{c}, \mathbf{r}\rangle$, where $\mathbf{z}$ contains the indices of the selected atoms out of $\mathbf{W}$, $\mathbf{c}$ are the corresponding coefficients and $\mathbf{r}$ is the residual. The set of all solutions is denoted as Ξ. Starting with a single trivial solution $\xi = \langle [\,], [\,], \mathbf{x}\rangle$, in every iteration each solution is extended with up to B atoms, selected by the function selectBestAtoms. In selectBestAtoms, we calculate $\mathbf{a} = \mathbf{W}^T\mathbf{r}$. Atoms with negative values in $\mathbf{a}$, and atoms which would make the prior probability (Eq. (10)) to zero, are discarded, where the prior probabilities are calculated according to the original dictionaries $\mathbf{W}^1$ and $\mathbf{W}^2$. When R is the number of remaining atoms, $\min(R, B)$ atoms with largest values in $\mathbf{a}$ are selected. The inner products and the indices of the selected atoms are returned in the vectors $\mathbf{a}^*$ and $\mathbf{a}_{idx}$. In lines 10-12, we perform NMF for the coefficient vector $\tilde{\mathbf{c}}$, which approximates Eq. (5).

Algorithm 3 Factorial SC - Inference

1: $\Xi \leftarrow \{\langle [\,], [\,], \mathbf{x}\rangle\}$
2: **for** l = 1:L **do**
3: $\quad\Xi^* \leftarrow \emptyset$
4: $\quad$**for** $\forall \xi \in \Xi$ **do**
5: $\quad\quad\langle \mathbf{z}, \mathbf{c}, \mathbf{r}\rangle \leftarrow \xi$
6: $\quad\quad[\mathbf{a}^*, \mathbf{a}_{idx}] = \text{selectBestAtoms}(B, \mathbf{W}, \mathbf{z}, \mathbf{r})$
7: $\quad\quad$**for** $b = 1 : |\mathbf{a}^*|$ **do**
8: $\quad\quad\quad\tilde{\mathbf{z}} \leftarrow [\mathbf{z}, \mathbf{a}_{idx}(b)]$
9: $\quad\quad\quad\tilde{\mathbf{c}} \leftarrow [\mathbf{c}, \mathbf{a}^*(b)]$
10: $\quad\quad\quad$**for** $j = 1 : J$ **do**
11: $\quad\quad\quad\quad\tilde{\mathbf{c}} \leftarrow \tilde{\mathbf{c}} \otimes \frac{(\mathbf{W}^T_{\tilde{\mathbf{z}}}\mathbf{x})}{(\mathbf{W}^T_{\tilde{\mathbf{z}}}\mathbf{W}_{\tilde{\mathbf{z}}}\tilde{\mathbf{c}})}$
12: $\quad\quad\quad$**end for**
13: $\quad\quad\quad\tilde{\mathbf{r}} \leftarrow \mathbf{x} - \mathbf{W}_{\tilde{\mathbf{z}}}\tilde{\mathbf{c}}$
14: $\quad\quad\quad\Xi^* \leftarrow \Xi^* \cup \langle \tilde{\mathbf{z}}, \tilde{\mathbf{c}}, \tilde{\mathbf{r}}\rangle$
15: $\quad\quad$**end for**
16: $\quad$**end for**
17: $\quad\Xi \leftarrow \Xi^*$
18: $\quad$**if** $l > T$ **then**
19: $\quad\quad$Prune Ξ to the B^T best solutions
20: $\quad$**end if**
21: **end for**

Continuing in this manner, the solution set comprises up to B^l solutions in iteration l. After $T + 1$ iterations, we start to prune the solution set to the B^T best solutions in every iteration, i.e. we select the B^T solutions with highest posterior (Eq. (11)), where the probabilities $p(\mathbf{z}^1)$ and $p(\mathbf{z}^2)$ are evaluated according to the original dictionaries. When the algorithm has stopped, we select the solution with maximal posterior out of the final solution set and build the coefficient matrix $\mathbf{H}$, which is split according to the original dictionaries: $\mathbf{H} =: \mathbf{H}^1 \cup \mathbf{H}^2$. The approximations of the source spectrograms are then given as $\hat{\mathbf{S}}^m = \mathbf{W}^m\,\mathbf{H}^m$. We calculate a mask for each source according to $\mathbf{M}^m = \frac{\hat{\mathbf{S}}^m}{\hat{\mathbf{S}}^1 + \hat{\mathbf{S}}^2},\ m \in \{1, 2\}$. Finally, approximations of the source signals are given by the inverse short term Fourier transform (ISTFT) of the masked mixture: $\hat{s}^m(t) = \text{ISTFT}(\mathbf{M}^m \otimes \bar{\mathbf{X}})$, where $\bar{\mathbf{X}}$ is the original complex mixture spectrogram.

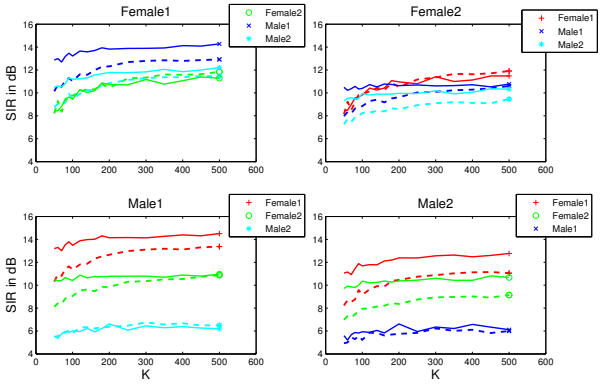

Figure 2: Separation results in SIR of factorial max-VQ (dashed) and factorial SC (solid). Each interfering speaker is marked with a specific color and marker.

5. Experiments

We selected two female and two male speakers from the SCSS database by Cooke et al. [12]. We refer to the speakers as *Female1*, *Female2*, *Male1*, and *Male2*. For each speaker we use 3 minutes of speech as training data, with a sampling frequency of 16kHz. The magnitude spectrograms were calculated using a 1024 samples hamming window and 512 samples overlap. Additionally, the logarithm was applied for factorial-max VQ. For both training methods (k-means and NMFℓ^0) 25 iterations were used. For NMFℓ^0, we used $J = 30$ NMF updates. For separation we selected 10 utterances of each speaker, which were not used for training. We mixed all possible combinations of files and speakers with a signal-to-interference ratio (SIR) of 0dB. This gives 100 mixture utterances for each speaker pair. We executed factorial-max VQ and factorial SC with varying dictionary (or code-book) sizes. For our method we also tried different values for the maximal allowed number of atoms L, where a value of $L = 3$ achieved good results. For inference (Algorithm 3) we used parameters $B = 4$ and $T = 2$. Figure 2 shows the mean achieved SIR after source separation for both methods as a function of the dictionary (code-book) size K. The SIR was calculated in the magnitude spectrogram domain in order to neglect phase distortions from resynthesis: $SIR = 10 \log_{10} \frac{\|\mathbf{S}^m\|_F^2}{\|\mathbf{S}^m - \hat{\mathbf{S}}^m\|_F^2}$, where $\mathbf{S}^m$ is the magnitude spectrogram of the m^{th} source, and $\hat{\mathbf{S}}^m$ is the spectrogram of its approximation. We see that our method performs better than factorial-max VQ in the different-gender case, and that the performance is approximately the same in the same-gender case. All experiments have been conducted on the same PC and the execution time needed for separation has been measured. Figure 3 compares the mean execution time for both methods as a function of the dictionary size. We see that the computational costs of factorial-max VQ increase dramatically with larger dictionaries, since in principle a full search over all codeword combinations is performed [4]. The computational effort for our method is dominated by the inference parameters B and T and increases only linear with K.

6. Conclusion

In this work we presented a probabilistic factorial sparse coder model for single channel source separation. In our model, the sources are modeled as the output of sparse coders plus Laplacian noise terms. To train the source-specific dictionaries and model parameters, we use non-negative matrix factorization with ℓ^0-sparseness constraints (NMFℓ^0). Further, we derived the posterior probability of the sparse-coder-atom selections for given mixture data. Since inference via exhaustive search is intractable, we restrict the set of considered solutions using a multi-hypotheses variant of matching pursuit. We compared separation performance to the factorial-max VQ system on the database provided by Cooke et al. [12]. Systematic separation experiments show the superior performance of the proposed algorithm in terms of signal-to-interference ratio. Finally, the algorithm is suitable for close to real time applications due to its low computational requirements.

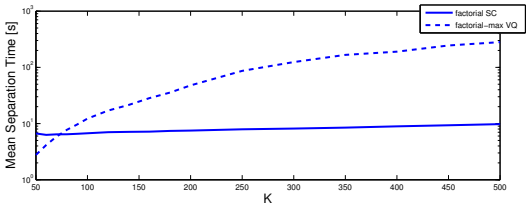

Figure 3: Mean execution time of factorial-max-VQ (dashed) and factorial SC (solid). The ordinate is in logarithmic scale.

Acknowledgement: This work was supported by the Austrian Science Fund (Project number: S10604-NB)

7. References

[1] A. Hyvärinen, J. Karhunen, and W. Oja, *Independent Component Analysis*. John Wiley & Sons, 2001.

[2] G. Brown and M. Cooke, "Computational auditory scene analysis," *Computer Speech and Language*, vol. 8, pp. 297–336, 1994.

[3] D. Wang and G. J. Brown, *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*, ser. IEEE Press. J. Wiley and Sons Ltd, 2006.

[4] S. Roweis, "Factorial models and refiltering for speech separation and denoising," in *EUROSPEECH*, 2003, pp. 1009–1012.

[5] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD and its non-negative variant for dictionary design," in *Proceedings of the SPIE conference, Curvelet, Directional, and Sparse Representations II*, vol. 5914, 2005, pp. 11.1–11.13.

[6] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximations," *Journal of Constructive Approximation*, vol. 13, pp. 57–98, 1997.

[7] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, 1993.

[8] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

[9] B. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," in *IEEE Trans. Signal Processing*, vol. 47, 1999, pp. 187–200.

[10] R. Peharz, "Single channel source separation using dictionary design methods for sparse coders," Master's thesis, Graz University of Technology, 2010.

[11] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[12] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *JASA*, vol. 120, pp. 2421–2424, 2006.