Master Thesis

Single Channel Source Separation using Dictionary Design Methods for Sparse Coders

Robert Peharz

Signal Processing and Speech Communication Lab
Technical University Graz
Head: Univ.-Prof. Dipl.-Ing. Dr.techn. Gernot Kubin

Assessor: Ass.-Prof. Dipl.-Ing. Dr.techn. Franz Pernkopf
Supervisors: Ass.-Prof. Yannis Stylianou, M. Sc., Ph. D.
Dipl.-Ing. Michael Stark

Graz, January 2010

Abstract

Single channel source separation aims to recover one or several source signals from a single mixture recording. Since we deal with at least 2 interfering sources, this problem is under-determined in any case. Bregman showed in [1] that the human auditory system uses various heuristics to separate the time-frequency plane of a perceived auditory scene, and that it reorganizes the resulting parts according to likely objects. Computational auditory scene analysis [2, 3, 4] aims to mimic this mechanism. Roweis [5] introduced the refiltering framework, where spectrogram masks are used to indicate the parts of the mixture spectrogram belonging to a specific source. Resynthesis of the source wave forms is achieved by modulating the original mixture phase onto the masking signals and applying the inverse Fourier transform. The challenging part is to estimate suited masking signals for each source. The factorial max vector quantization (max-VQ) system [6] models the source spectrograms with independent vector quantizers, and estimates the most probable states for each source given the mixture data. The corresponding code words of the vector quantizers give an approximation of the source spectrograms, which can be used to estimate the masking signals.

The K-SVD algorithm [7] was proposed for the design of overcomplete dictionaries for sparse coders. On the other hand, this algorithm can be seen as generalization of k-means, the standard training algorithm for vector quantizers. In this thesis we aim to extend the factorial max-VQ system by replacing k-means with a more flexible and more expressive training method. We propose a new algorithm which combines nonnegative sparse coding with nonnegative matrix factorization (NMF) [8], which we call NMF with ℓ^0 constraints. We develop a probabilistic framework for single channel source separation and compare our system to factorial max-VQ in systematic experiments using data from the database by Cooke [9]. These experiments confirm that our system performs on average better than factorial max-VQ in terms of signal-to-interference ratio. Furthermore, the proposed method needs a much lower computational effort, so that it can be executed up to 15 times faster than the baseline system. Additionally, we apply our algorithm to real-world mixture data recorded from various TV broadcasts [10].

Kurzfassung

Single Channel Source Separation versucht eine oder mehrere Quellen aus einer einkanaligen Mischung zu extrahieren. Dieses Problem ist unterdeterminiert, da mindestens 2 Quellen miteinander interferieren. Bregman zeigte in [1], dass das menschliche Gehör mehrere Heuristiken verwendet um die Zeit-Frequenz Darstellung einer akustischen Wahrnehmung in Teile zu zerlegen und diese dann entsprechend plausibler Objekte reorganisiert. Computational Auditory Scene Analysis [2, 3, 4] versucht diesen Mechanismus zu imitieren. Roweis [5] führte die Methode des Refiltering ein, bei der Spektrogrammmasken die Teile einer bestimmten Quelle im Spektrogramm der Mischung markieren. Mit Hilfe dieser Masken können die Quellsignale resynthetisiert werden, indem die originale Phase des Mixturspektrogramms der Maske aufmoduliert wird, und diese einer inversen Fourier Transformation unterzogen wird. Die Herausforderung dabei ist es, geeignete Maskensignale für jede Quelle zu schätzen. Das factorial max Vector Quantization (max-VQ) System [6] modelliert die Quellspektrogramme mit unabhängigen Vektor Kodierern und schätzt die wahrscheinlichsten Zustände für jede Quelle bei gegebenen Mischung. Die dazugehörigen Kodevektoren ergeben eine Approximation der Quellspektrogramme, mit denen die Masken geschätzt werden können.

Der K-SVD Algorithmus [7] wurde zum Design von überkompletten Wörterbüchern für Sparse Coder entwickelt. Andererseits kann dieser Algorithmus als Generalisierung von k-means gesehen werden, dem Standardtrainingsalgorithmus für Vektor Kodierer. In dieser Masterarbeit versuchen wir das factorial max-VQ System zu erweitern, indem wir k-means mit einer flexibleren und ausdrucksstärkeren Methode ersetzen. Wir präsentieren einen neuen Algorithmus, der eine Kombination von nicht-negativem Sparse Coding und nicht-negativer Matrix Faktorisierung (NMF) [8] darstellt. Wir nennen diesen neuen Algorithmus nicht-negative Matrix Faktorisierung mit ℓ^0 constraints. Wir entwickeln ein probabilistisches Modell für Single Channel Source Separation und vergleichen unsere Methode mit factorial max-VQ in systematischen Experimenten, wobei wir die Datenbank von Cooke [9] verwenden. Diese Experimente zeigen, dass unser System im Durchschnitt ein höheres Signal-Interferenz Verhältnis als factorial max-VQ erreicht. Weiteres ist der Rechenaufwand für die vorgeschlagene Methode wesentlich niedriger, so dass sie bis zu 15 Mal schneller als das Vergleichssystem ausgefhrt werden kann. Außerdem wenden wir unseren Algorithmus auf realistische Mischungen von verschiedenen TV Sendungen an [10].

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Date:

Signature:

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am:

Unterschrift:

Acknowledgement

This thesis was written during my Erasmus semester in Greece, at the Computer Science Department of the University of Crete, in cooperation with the Signal Processing and Speech Communication Lab, Graz, University of Technology.

I would especially like to thank Yannis Stylianou and his group for supervising and supporting my work, for many interesting discussions and taverna visits. Also many thanks to Franz Pernkopf, my supervisor and assessor in Graz. Special thanks go to Michael Stark, who provided me with data and various code parts, and whose experience in Single Channel Source Separation was very helpful for my work. He also suggested and organized the automatic speech recognition experiment in Section (4.4).

Further I like to thank George Murry and my father Herbert Peharz for proofreading my thesis.

Contents

Chapter 1

Introduction

Humans constantly receive a large amount of information via their senses. The ability to cope with this stream of information is essential for us to survive and to orient ourselves in our environment. Although we interpret the information we gain with apparent ease, this task is not easy to be automatized and implemented in a computer.

One task the human brain solves during the processing of incoming information, is to split the sensory input into meaningful parts and to reorganize these according to objects. More precisely, it solves the so called source separation problem, in the auditory domain also known as the "Cocktail Party Problem" [11].

Humans show great skill in perceiving a certain sound source out of a mixture. We can hear a specific voice in an environment like a noisy cocktail party, although the surrounding noise may be much louder than the voice we are concentrating on. This outstanding ability has motivated many different approaches to implement effective computational systems which are able to solve the source separation problem. These systems can be used as a front end for automatic speech recognizers, since their performance usually drops dramatically when the target voice is interfered with noise, speech or other sounds. Other applications for source separation systems are audio processing, automatic music transcription systems and smart hearing aids.

In general, the problem of source separation is to extract one or several sources from a set of n recordings, where the source signals are mixed in a different way for each recording. Figure (1.1) illustrates this setup for $n = 3$. If we have only one mixture signal ($n = 1$), we speak about single channel source separation (SCSS). Note that since we have at least 2 source signals, this case is necessarily underdetermined, and several systems proposed so far can not be applied. However, in various applications we do not have several recordings of an auditory scene, like in telephony, or we are not willing to make the effort with more than one channel. This fact should motivate us to develop methods which can treat the SCSS task in a more satisfying manner.

1.1 Scope of Research

In this thesis we extend the factorial max-vector quantization (VQ) model for SCSS introduced by Roweis [6]. This system models the magnitude-log spectrogram of the sources with the output of vector quantizers plus Gaussian noise. For SCSS, the most probable

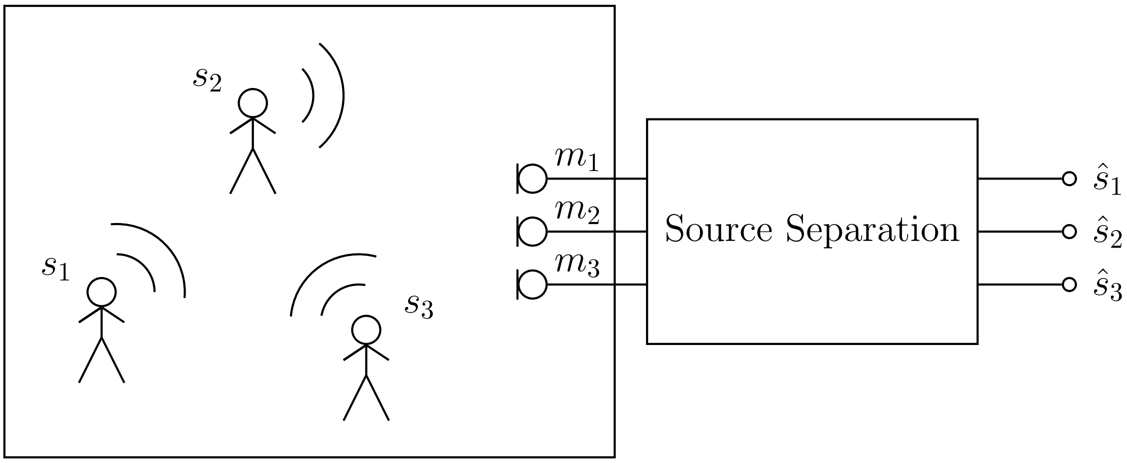


Figure 1.1: Principal setup for source separation.

code words for each VQ are inferred, which represent approximations of the source spectrograms. These approximations can be used to calculate so called binary masks, which again can be used for the resynthesis of the source waveforms. However, the codebooks of the VQ have to be trained for source specific data, where the well known k-means algorithm is used.

Aharon et al. [12] proposed the K-SVD algorithm (K singular value decompositions) for the design of overcomplete dictionaries for sparse coders, which was successfully applied for data compression and denoising. K-SVD can be seen as a generalized version of k-means and works in an *expectation-maximization* like manner. Therefore, to extend the factorial max-VQ using K-SVD instead of k-means would seem to be promising. Aharon et al. [12] provided also a nonnegative version of K-SVD, which is more suitable for this task, since factorial max-VQ operates with magnitude-log spectrograms, i.e. nonnegative data. However, in this thesis we introduce a simpler and more effective version of nonnegative K-SVD, by replacing the M-step with nonnegative matrix factorization (NMF), as introduced by Lee and Seung [8]. This new algorithm is called nonnegative matrix factorization with ℓ^0 constraints (NMFℓ^0). For the E-step, which consists of the application of a sparse coder to the training data, we adapt the orthogonal matching pursuit (OMP) [13] in order to satisfy nonnegativity constraints. We call this adapted version of OMP nonnegative matching pursuit (NMP). Additionally, we introduce a generalized algorithm which represents an intermediate solution between the fast but greedy NMP and the optimal but intractable full search. This generalized algorithm is called beam-search NMP (BS-NMP).

Analog to factorial max-VQ, we define a probabilistic framework which models the sources with sparse coders and which we call factorial sparse coder model (factorial SC). Since it is intractable to search the optimal solution for this model, we restrict the search space to a sub-set of plausible solutions.

We show in systematic experiments with synthetic mixture data [9], that the proposed algorithm performs better than max-VQ in terms of the mean signal-to-interference ratio. Additionally, the computational effort of the new method is much lower than for the baseline system, which allows close to real-time applications. Finally, we apply factorial

SC to real-world mixture data recorded from various TV broadcasts [10]. To evaluate the separation quality, we use informal listening tests and compare the results of an automatic speech recognizer (ASR), one time applied to the original mixture and the other time applied to the separated signals. While we can state due to the listening tests that our algorithm clearly extracts the target source, the ASR results significantly increase only for one target out of four.

1.2 Organization of the Thesis

In chapter 2, we review several approaches for source separation. We discuss independent component analysis (ICA), the standard algorithm for general source separation. We argue that ICA and sparsity are related concepts, and that a sparse code is a useful representation for cognitive systems. The baseline system of our work, factorial max-VQ, is also discussed.

In chapter 3, we describe K-SVD in detail. We introduce the alternative algorithm for nonnegative K-SVD, which we call nonnegative matrix factorization with ℓ^0 constraints (NMFℓ^0). The adapted version of OMP, nonnegative matching pursuit (NMP), and its generalized version, beam-search NMP, are also discussed.

In chapter 4, we apply K-SVD and NMFℓ^0 to the SCSS problem by using a simple straight forward solution. We perform a best case analysis, in order to estimate an upper performance bound which can be achieved when K-SVD or respectively NMFℓ^0 dictionaries are used in a refiltering approach. Further we define the factorial sparse coder model, and use the BS-NMP algorithm in order define a plausible restricted search space, and to find the optimal solution within this restricted space.

Finally, chapter 5 concludes this thesis.

Chapter 2

Related Work

In this chapter we review some source separation systems, where in general we can distinguish two main approaches. The first approach relies on statistical properties of the source signals and is known as independent component analysis (ICA). In section 2.1 we describe ICA in its classical definition as the solution for an unmixing problem. We also point out the relationship between ICA and sparse coding, and argue that sparseness is a useful concept for cognitive information processing systems. The second approach is known as computational auditory scene analysis (CASA), which is is briefly reviewed in section 2.3. Generally, CASA tries to mimic the ability of humans to separate sound sources by applying a so called auditory scene analysis (ASA) [1]. Section 2.4 introduces the factorial max vector quantization (max-VQ) system by Roweis [6], which tries to combine the main ideas of ICA and CASA. On the one hand a technique called refiltering is basically used by all CASA systems and on the other hand statistical methods are applied for the estimation of so called masking signals. In section 2.5 we discuss Nonnegative Matrix Factorization (NMF) [8] and its application to the SCSS problem.

2.1 Independent Component Analysis

Independent component analysis (ICA) [14] is the classical approach to the source separation problem. In its standard definition, ICA needs at least as many mixture signals as source signals. Although under-determined versions of ICA exist [14], it can not be applied in the monophonic case.

Consider an auditory scene with n sources and n microphones. According to the laws of physics, the signal waveforms emitted by the sources are mixed in an additive way. Due to different distances between sources and microphones, and due to the specific geometric constellation in the given scene, the sources will be mixed with different gain factors. When we denote the i^{th} signal with $s_i(t)$ $(1 \leq i \leq n)$ and the j^{th} mixture with $m_j(t)$ $(1 \leq j \leq n)$, then the mixture model has the following form:

$$m_j(t) = a_{j,1}s_1(t) + a_{j,2}s_2(t) + \ldots + a_{j,n}s_n(t) \ , 1 \leq j \leq n. \tag{2.1}$$

The variable t denotes time and $a_{j,i}$ are the gain factors for the i^{th} signal and the j^{th} mixture. Note that in a realistic setup the assumption that the mixtures are generated by a mere linear combination of the source signals does not hold. Usually we have to consider delay factors due to different distances and the impulse responses of the surrounding area. However, for simplicity's sake let us consider this simple linear mixture model. In matrix notation, the model according to Eq. (2.1) takes the compact form

$$\mathbf{m}(t) = \mathbf{A}\,\mathbf{s}(t). \tag{2.2}$$

The source and the mixture signals are arranged into vectors **s** and **m** respectively, and the gain factors are combined to the mixing matrix **A**. We see now clearly how to solve the problem. If we knew the mixing matrix **A**, and if it was invertible, we simply have to multiply the inverse mixing matrix with the mixture vector in order to regain the source signals:

$$\mathbf{s}(t) = \mathbf{A}^{-1}\,\mathbf{m}(t). \tag{2.3}$$

However, usually we do not know **A**. The key idea in ICA is that we can usually that the sources are statistically independent. Therefore we try to find a demixing matrix so that the demixed signals are as independent as possible. However, it is not easy to measure independence. The standard definition of statistical independence is that the joint probability density function (pdf) of a random vector factorizes into the product of the marginal pdfs of the individual random variables. In order to use this definition we would have to estimate the joint pdf and the product of the marginal pdfs of the demixed signals and to compare these via some measure like the Kullback-Leibler divergence.

The central limit theorem provides an easier method to measure statistical independence. The pdf of the sum of independent random variables with nongaussian pdfs is closer to a Gaussian distribution than the individual pdfs. Therefore we can use the dissimilarity between the pdf of the demixed signals and a Gaussian distribution in order to measure independence. The prerequisite for this approach is that the original signals are distributed according to nongaussian pdfs. The Gaussian distribution is fully determined by its mean value and its variance, while all higher order cumulants are zero. It is also the *only* distribution whose higher order cumulants are all zero. Therefore we can aim to maximize the absolute value of higher order cumulants to achieve nongaussianity and hence independence.

The classical choice for ICA is to maximize the fourth order cumulant, the kurtosis. The kurtosis corresponds to the "peakyness" of a pdf. Pdfs with large positive kurtosis are called super-Gaussian and are concentrated in a peak around the mean value with "heavy tails", what means that they fall off slower than the Gaussian distribution for values which are far away from the mean. On the other hand, pdfs with negative kurtosis are called sub-Gaussian and are more similar to the uniform distribution.

In order to increase the statistical independence of the unmixed signals, we can use a gradient ascent method to maximize the absolute value of the kurtosis. The most basic ICA algorithm therefore consists of:

1. Remove the mean of the mixed signal.

2. Prewhite the mixed signals, i.e. decorrelate the mixture signals and normalize to unit variance.

3. Find a demixing matrix that maximizes the absolute value of the kurtosis of the demixed signals, using a gradient ascend method.

Bell and Sejnowski [15] developed a new view on ICA. They generalized the infomax principle by Linsker [16, 17] to neural networks with nonlinear activation functions. The infomax approach intents to maximize the information flow from input to output (i.e. the mutual information) in a network by tuning the network weights. They showed that information maximization between input and output generally yields into a reduction of the mutual information between the outputs. Hence, the statistical independence of the outputs is increased.

2.2 ICA and Sparsity

Bell and Sejnowski applied their algorithm to patches of natural images [18] and found that the independent components are similar to observed responses of simple cells in the visual cortex. These responses can be described as localized, oriented and similar to Gabor filters. On the other hand, Olshausen and Field [19] drew parallels between the observed coding in the primary visual cortex and a sparse coding strategy. Their sparse coding algorithm turned out to be very similar to the ICA algorithm by Bell and Sejnowski and produced similar results. This shows that ICA and sparse coding are familiar concepts, which can be illustrated by the following argument. ICA tries to transform data so that the output has higher statistical independence, where we can use the kurtosis as an independence measure. Pdfs with high kurtosis are peaky around their mean value. Since a sparse code tries to use only few significantly active elements to code the data, the pdf of the sparse output will naturally have a peak around zero and therefore have a high kurtosis. Thus the output of a sparse coder will usually be more statistically independent than the input. However, ICA using kurtosis as an independence measure will only produce a sparse code, if the kurtosis is actually *maximized.* In ICA we usually aim to maximize the absolute value of the kurtosis, which allows the kurtosis to also be negative. However, in many cases we deal with signals with positive kurtosis, like e.g. speech signals.

Field [20] proposed that a general strategy for early sensory input could be a sparse distributed code. He distinguished between sparse distributed codes and compact codes. A compact code has the goal to describe multidimensional input with as few basis vectors (cells) as possible. These vectors can be adapted to certain data by minimizing the reconstruction loss. The result is a dimensionality reduction and a compression. In contrast, a sparse distributed code does not aim to reduce the dimensionality, but rather to minimize the number of *active* cells. The difference between compact coding and sparse-distributed coding is illustrated in figure (2.1). Sparse representation of sensory input has several advantages. The authors in [21] and [22] show, that an associative memory network has significantly higher capacity when its input shows a sparse structure. Foeldiak [23] argues, that the training of these networks becomes slow for distributed input, and that the resulting learning rules are more complicated. Moreover, it is plausible that a cognitive system can more easily handle input in sparse form. When we interpret the basis vectors as features, a sparse activation implies that a certain feature is inactive most of the time. On the other hand, if a feature is active for specific input, this means that we registered an important event connected to this feature.

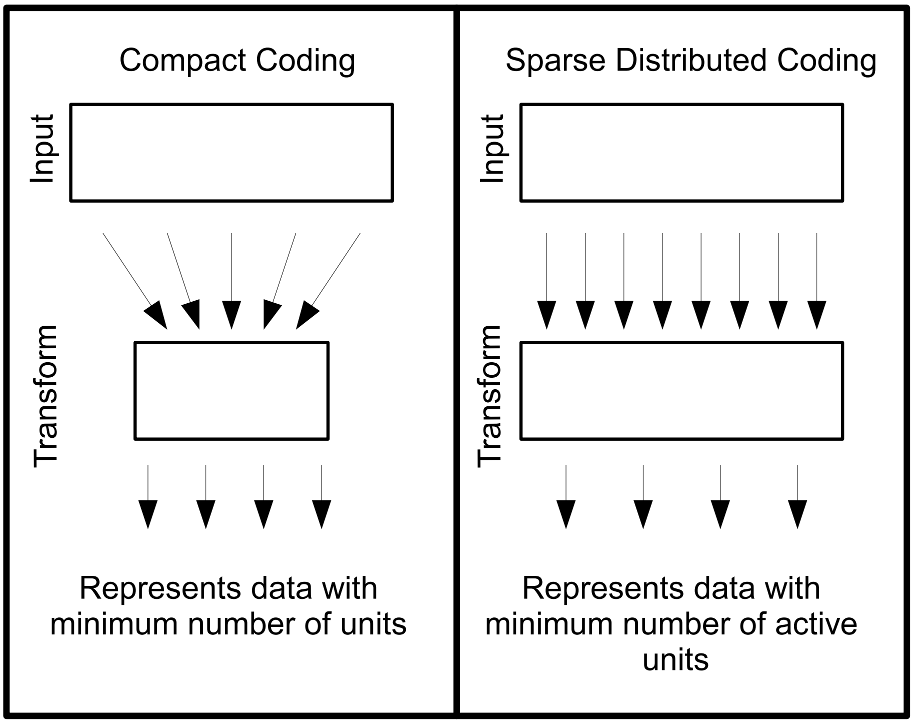

Compact Coding
Input
Transform
Represents data with minimum number of units
Sparse Distributed Coding
Input
Transform
Represents data with minimum number of active units

Figure 2.1: Compact coding versus sparse-distributed coding [20].

2.3 Computational Auditory Scene Analysis

Bregman [1] found that the human brain seems to use various heuristics in order to perform the source separation task. He called this process auditory scene analysis (ASA), in parallel to scene analysis in computer vision. The cochlea in the inner ear provides a short term frequency analysis of the incoming sound. The parts of this time-frequency representation are organized by the brain into so called streams, where several grouping rules are used, like common on- and off-set of energy in different frequency bands, harmonic stacking and co-modulations of quasi-sinusoids. The work by Brown and Cooke [2] was the first attempt to design a sophisticated system which takes use of the physiological and psychological findings about the human auditory system. Wang and Brown developed in [3] a neurologically inspired oscillatory correlation model for the source separation task.

2.4 Factorial max-VQ model

Roweis [5] recognized that the grouping rules of the CASA systems proposed so far are more or less hand designed. He proposed to build explicit models by applying machine learning techniques to training data of the sources. In the separation step, these models are used to find the components corresponding to the sources. Roweis introduced the general framework of refiltering, where the first step is to split the mixture into sub bands using a filter bank. By weighting each channel with a nonstationary masking weight between 0 and 1, and by recombinating the weighted channels, a certain source signal can be resynthesized. In more detail, if $b_i(t)$ denotes the i^{th} mixture channel and $a_i(t)$ are the weights to choose, a certain source signal $s(t)$ can be recovered by

$$s(t) = \sum_i a_i(t) b_i(t) \tag{2.4}$$

Two signals rarely interfere in both time and frequency, therefore this approach is very promising. In contrast to ICA, where several mixtures are reweighted and added, the refiltering approach reweights and adds the sub bands of a single mixture. The short time Fourier transform (STFT) is equivalent to a filter bank with linear equidistantly spaced center frequencies. Refiltering can therefore be performed by reweighting the mixture spectrogram with the masking signals, applying the inverse Fourier transform to the columns, and an overlap-add procedure. The remaining and challenging task is to estimate the correct masking signals $a_i(t)$ in order to extract a desired source. Roweis made the simplification that the masking signals are binary (i.e. 0 or 1) and stationary in each analysis frame. Each spectrogram bin is therefore assigned to only one source. For a small number of sources however, this simplification still allows good results. The factorial max-VQ system [6] models the magnitude-log spectrograms of the sources as the output of vector quantizers. Further, the magnitude-log spectrogram of the mixture is approximated by the element-wise maximum of the magnitude-log spectrograms of the sources, plus a Gaussian noise term. In this system, the output in a certain time frame is assumed to be independent from all other frames. Therefore, source separation can be performed for each spectrogram column separately. In [5], the factorial-max HMM model was proposed which also considered time dependencies using hidden Markov models (HMM). However, the factorial-VQ model performs almost as well as this system, while needing much less computational effort.

When we assume that M sources are present in an auditory scene, the factorial-max VQ model consist of M vector quantizers, each with K_M codebook entries, where the k^{th} code word of the m^{th} VQ is denoted by $\mathbf{v}_m^k$. The latent variables $z_m \in \{1, \ldots, K_m\}$, $1 \leq m \leq M$ hold the indices of the codewords which are selected by the VQ. The probabilities of the code words are denoted by π_m^k and are assumed to be fixed:

$$\pi_m^k = p(z_m = k), \quad 1 \leq m \leq M, \ 1 \leq k \leq K_m. \tag{2.5}$$

The joint probability of the outputs is given as $p(\mathbf{z}) = \prod_m p(z_m)$, where $\mathbf{z} = (z_1, \ldots, z_m)$. Given the codebook selections $\mathbf{z}$, the index a_d of the VQ which has the maximal output in the d^{th} dimension is

$$a_d = \arg\max_m \left(v_m^{z_m}\right)_d . \tag{2.6}$$

Hence, the d^{th} entry of the mixture spectrogram x_d is distributed according to

$$p(x_d | a_d, \mathbf{v}, \Sigma^2) = \mathcal{N}(x_d | v_{a_d d}^{z_{a_d}}, \Sigma_{a_d d}^2), \tag{2.7}$$

where $\mathcal{N}(\cdot|\mu, \sigma^2)$ is the Gaussian distribution with mean value μ and variance σ^2. The value $v_{a_d d}^{z_{a_d}}$ is the maximum of the outputs of the sparse coders in the d^{th} dimension and Σ_{md}^2 are noise variances, which are shared between the code words of one VQ. The entries of the mixture $\mathbf{x}$ are assumed to be independent of each other, so that the probability of

$\mathbf{x}$ given the codebook selections $\mathbf{z}$ can be written as

$$p(\mathbf{x}|\mathbf{z},\mathbf{v},\Sigma^2)) = \prod_{d=1}^{D} p(x_d|a_d,\mathbf{v},\Sigma^2), \tag{2.8}$$

where D is the number of spectrogram frequency bins. The marginal probability of $\mathbf{x}$ is given as

$$p(\mathbf{x}|\mathbf{v},\Sigma,\pi) = \sum_{\mathbf{z}} p(\mathbf{z})p(\mathbf{x}|\mathbf{z},\mathbf{v},\Sigma^2). \tag{2.9}$$

In order to train the code books for the VQ model, k-means is applied to the magnitude-log spectrograms of source specific training data. The residual error of k-means can be used to estimate the noise variances Σ^2_{md} and the prior probabilities π^k_m can be estimated by counting code word appearances.

For source separation, following a maximum *a posteriori* (MAP) approach, we are interested in a single $\mathbf{z}$ given the mixture $\mathbf{x}$, which maximizes the summand in Eq. (2.9), since

$$p(\mathbf{z}|\mathbf{x}) \propto p(\mathbf{z})p(\mathbf{x}|\mathbf{z},\mathbf{v},\Sigma^2). \tag{2.10}$$

To find the optimal $\mathbf{z}$ in principle a full search has to be performed, which has a computational complexity of $O(K^M)$ (assuming that $K_m = K$, $\forall m$). Roweis proposed a branch and bound technique in order to alleviate this problem. He defined an upper bound for the posterior probability for each code word, independent from the state of all other VQ. When the upper bound of a code word is below the exact posterior of an already considered combination of code words, then this code word does not have to be considered any more.

When the selection $\mathbf{z}$ with the highest posterior is determined, then the code book entry $\mathbf{v}^{z_m}_m$ represents an estimation of the log-magnitude spectrogram column of the m^{th} source. Repeating this process for every time frame, we get approximations of the whole source log-magnitude spectrograms. Further, we can use these approximations to estimate a set of binary or continuous masks, one for each source. For the continuous masks it is necessary to invert the logarithm in order to obtain the magnitude spectrograms in the linear domain. Let $\hat{\mathbf{S}}^m_{dn}$ denote the estimation of the linear magnitude spectrogram of the m^{th} source, where d corresponds to the frequency bin and n to the time index. The binary mask (BM) for each source is given as

$$\mathbf{BM}^m_{dn} = \begin{cases} 1 & \text{if } \hat{\mathbf{S}}^m_{dn} > \hat{\mathbf{S}}^l_{dn} \quad \forall l \neq m \\ 0 & \text{otherwise} \end{cases}. \tag{2.11}$$

The continuous mask (CM) for each source is given as

$$\mathbf{CM}^m_{dn} = \frac{\hat{\mathbf{S}}^m_{dn}}{\sum_l \hat{\mathbf{S}}^l_{dn}}. \tag{2.12}$$

2.5 Nonnegative Matrix Factorization

Lee and Seung [8, 24] proposed nonnegative matrix factorization (NMF) in order to represent a nonnegative matrix $\mathbf{X}$ as a matrix product of nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$:

$$\mathbf{X} \approx \mathbf{W}\ \mathbf{H}. \tag{2.13}$$

When $\mathbf{X}$ is a $D \times N$ matrix, then $\mathbf{W}$ and $\mathbf{H}$ have the dimensions $D \times R$ and $R \times N$, respectively, where the inner dimension (approximation level) R has to be chosen by the user. When we interpret the columns of $\mathbf{X}$ as data samples, then the columns $\mathbf{w}_i$ ($i = 1 \dots R$) of $\mathbf{W}$ can be interpreted as bases vectors and the elements of $\mathbf{H}$ as corresponding weight coefficients, since

$$\mathbf{x}_j \approx \sum_{i=1}^{R} h_{ij}\mathbf{w}_i \quad 1 \leq j \leq N. \tag{2.14}$$

Since the coefficients are nonnegative, each data point is represented by a mere addition of nonnegative bases vectors, without allowing subtractions. Eq. (2.13) implies that we have to minimize a distance measure between $\mathbf{X}$ and its reconstruction $\mathbf{W}\,\mathbf{H}$. Lee and Seung showed in [8] that the Euclidean distance is non increasing under the update rules

$$\begin{aligned} \mathbf{H}_{a\mu} &\leftarrow \mathbf{H}_{a\mu} \otimes \frac{(\mathbf{W}^T\mathbf{X})_{a\mu}}{(\mathbf{W}^T\mathbf{W}\,\mathbf{H})_{a\mu}}, \\ \mathbf{W}_{ia} &\leftarrow \mathbf{W}_{ia} \otimes \frac{(\mathbf{X}\,\mathbf{H}^T)_{ia}}{(\mathbf{W}\,\mathbf{H}\,\mathbf{H}^T)_{ia}}. \end{aligned} \tag{2.15}$$

where $\otimes$ and / denote element wise product (Hadamard product) and element wise division, respectively. Similarly as for the Euclidean distance, they showed that the Kullback-Leibler divergence between $\mathbf{X}$ and $\mathbf{W}\,\mathbf{H}$ is non increasing under the following update rules.

$$\begin{aligned} \mathbf{H}_{a\mu} &\leftarrow \mathbf{H}_{a\mu} \otimes \frac{\sum_i \mathbf{W}_{ia}\,\mathbf{X}_{i\mu}/(\mathbf{W}\,\mathbf{H})_{i\mu}}{\sum_k \mathbf{W}_{ka}}, \\ \mathbf{W}_{ia} &\leftarrow \mathbf{W}_{ia} \otimes \frac{\sum_\mu \mathbf{H}_{a\mu}\,\mathbf{X}_{i\mu}/(\mathbf{W}\,\mathbf{H})_{i\mu}}{\sum_\nu \mathbf{H}_{a\nu}}. \end{aligned} \tag{2.16}$$

By inspection of the equations (2.15) and (2.16) it is clear, that they will not violate the nonnegativity constraint, given that $\mathbf{X}$ and the initial matrices $\mathbf{W}$ and $\mathbf{H}$ are nonnegative.

Hoyer [25] proposed an NMF version with additional sparsity constraints. In order to define sparsity, he used the following measure for an arbitrary vector $\mathbf{x}$ with n elements:

$$\text{sparseness}(\mathbf{x}) = \frac{\sqrt{n} - L_1(\mathbf{x})/L_2(\mathbf{x})}{\sqrt{n} - 1}, \tag{2.17}$$

where $L_1(\cdot)$ and $L_2(\cdot)$ denote the ℓ^1 norm and ℓ^2 norm, respectively. Indeed, this function will be 0 only when the $\mathbf{x}$ are all equal and nonzero, and it will be 1 when all entries except one are 0. For all other vectors this function smoothly interpolates between these extreme cases. Hoyer provided a gradient descend method to reduce $\|\mathbf{X} - \mathbf{W}\,\mathbf{H}\|_F^2$, where the columns of $\mathbf{W}$ or $\mathbf{H}$ (or both) are constrained to have the sparseness according to Eq. (2.17). $\|\cdot\|_F^2$ denotes the Frobenius norm. Virtanen [26] applied sparse NMF for an automatic transcription system.

Analog to Roweis's system, we can use NMF for SCSS in a two stage approach. Let us assume M interfering sources. In the training stage we take for each source the magnitude spectrogram $\mathbf{S}^m$ of source specific training data and remove the redundant frequency bins. We train bases sets $\mathbf{W}^m$ for each source by applying the NMF update rules to $\mathbf{S}^m$, where $\mathbf{W}^m$ and $\mathbf{H}^m$ are initialized randomly.

In the separation step we build the concatenation $\mathbf{W}$ of all bases matrices, i.e. $\mathbf{W} = \mathbf{W}^1 \cup \; \mathbf{W}^2 \cdots \cup \; \mathbf{W}^M$. Starting from a randomly initialized coefficient matrix $\mathbf{H}$, we apply the NMF update rule for $\mathbf{H}$ to the mixture spectrogram $\mathbf{X}$, while keeping $\mathbf{W}$ fixed. The resulting coefficient matrix can be split into parts $\mathbf{H}^1, \; \mathbf{H}^2, \; \ldots, \; \mathbf{H}^M$, according to the original bases sets. Finally, we can calculate an approximation $\hat{\mathbf{S}}^m$ of the m^{th} source spectrogram as $\hat{\mathbf{S}}^m = \mathbf{W}^m \, \mathbf{H}^m$. With these approximations we can calculate binary or continuous masks according to Eq. (2.11) and Eq. (2.12), respectively, and resynthesize the source waveforms.

Chapter 3

The K-SVD algorithm

The K-SVD algorithm was proposed by Aharon et al. [7] as a method for the design of overcomplete dictionaries for sparse coders. A sparse coder approximates a given signal $\mathbf{x}$ with a linear combination of maximal L so called signal atoms out of a dictionary containing K atoms. Formally, this means

$$\mathbf{x} \approx \sum_{i=1}^{L} h_{z_i} \mathbf{w}_{z_i}, \tag{3.1}$$

where $\mathbf{w}_k$ is the k^{th} signal atom, h_k is the corresponding coefficient and $\mathbf{z} = (z_1, z_2, \ldots, z_L)$ denote hidden variables which hold the indices of the atoms which are used to approximate $\mathbf{x}$, i.e. $1 \leq z_i \leq K, \forall i$ and $z_i \neq z_j,$ for $i \neq j$. We can reformulate Eq. (3.1) as

$$\mathbf{x} \approx \mathbf{W}\ \mathbf{h}, \quad \text{s.t. } L_0(\mathbf{h}) \leq L \tag{3.2}$$

where L_0 denotes the ℓ^0 norm and the signal atoms are organized in the columns of the dictionary matrix $\mathbf{W}$. The vector $\mathbf{h}$ holds the coefficient h_{z_i} in the ${z_i}^{\text{th}}$ entry and has zeros elsewhere. Thus it encodes both the coefficients h_{z_i} and the hidden variables $\mathbf{z}$.

To represent a signal in a sparse way has many advantages and applications, like data compression, denoising, feature extraction and more. Also, we argued in section 2.2 that sparse coding is related to ICA and that a sparse distributed code has several advantages for information processing systems. To provide a suitable dictionary to a sparse coder is crucial for the result. Many predefined dictionaries have been proposed for various signal classes like the overcomplete Wavelet dictionary or the Haar dictionary [27]. The next step is to design a dictionary for a specific signal class, given as a set of training examples. Let us assume N given training signals of length D which are arranged in the columns of the data matrix $\mathbf{X}$. We can formally define the dictionary design task as minimizing the objective

$$E_L = \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 \quad \text{s.t. } L_0(\mathbf{h}_i) \leq L, \tag{3.3}$$

where $\mathbf{h}_i$ is the i^{th} column of $\mathbf{H}$, which is accordingly the coefficient vector for the i^{th} training signal $\mathbf{x}_i$. An alternative objective proposed in [7] is the average number of used dictionary atoms while maintaining a given error bound, i.e.

$$L_\epsilon = \langle L_0(\mathbf{h}_i) \rangle \quad \text{s.t. } \|\mathbf{X} - \mathbf{W}\mathbf{H}\|_F^2 \leq \epsilon, \tag{3.4}$$

where $\langle \cdot \rangle$ denotes the average operator.

Throughout this thesis we will only use the definition related to Eq. (3.3). Firstly, because also the authors of K-SVD primary develop their ideas according to this definition. Secondly, the K-SVD algorithm as proposed in [7] does not guarantee to solve Eq. (3.4). In some rare cases it does indeed increase the number of used dictionary atoms. Thirdly, it is the first approach Eq. (3.3) which can be seen as a generalization of k-means, and which is therefore suitable to extent the system by Roweis [6]. Indeed, when we set the parameter L to 1, which means that maximal one atom is allowed to be used for each training example, and, if the value of this coefficient is constrained to be 1 ($h_{z_1} = 1$), then an algorithm which minimizes Eq. (3.3) reproduces per definition the results of k-means.

Similar as k-means, K-SVD works in an *expectation-maximization* (EM) like manner. The E-step in k-means is the assignment of each data point to one of K cluster centers. In K-SVD this step is generalized in order to find an assignment to up to L dictionary atoms *and* the respective coefficients. Regarding Eq. (3.3), this means to find an assignment vector $\mathbf{z}$ and the corresponding coefficients $h_{z_i}, 1 \leq i \leq L$, so that $\mathbf{x}$ is optimally approximated. Therefore, the E-step of K-SVD consists of running a sparse coder for each training example, while holding the dictionary fixed. In the next section we will review the problem of sparse coding and describe the orthogonal matching pursuit algorithm, which is primarily used in K-SVD.

The M-step on the other side aims to improve the dictionary using the result of the sparse coding step. Relating to Eq. (3.3), this means to optimize the dictionary entries $\mathbf{w}_k$ and their corresponding coefficients, while holding the assignment vector $\mathbf{z}$ fixed. In k-means this is achieved by updating the cluster centers with the mean vectors of the assigned data clusters. In K-SVD however, this step applies the calculation of a singular value decomposition (SVD) of an error matrix for each of the K atoms, hence the name K-SVD. In section 3.2, we describe this step in detail.

There are several ways to initialize the dictionary. Possible choices are prespecified dictionaries, random dictionaries or randomly picked signals out of the set of training signals.

3.1 Sparse Coding

Formally, the problem to solve is to minimize the objective given in Eq. (3.3) with respect to $\mathbf{H}$ only, while keeping $\mathbf{W}$ fixed. Unfortunately, the sparse coding problem is known to be NP-hard [28], since all possible $\binom{L}{K}$ combinations had to be considered in order to guarantee the optimal solution. Note that the actual challenging task is to *select* the atoms which are able to approximate a certain data sample best. If we had this information, Eq. (3.3) can be minimized by a least squares (LS) approximation using the selected atoms.

Since it is unfeasible to find the optimal solution, we have to use approximative methods. Simple and yet effective algorithms are matching pursuit (MP) [29] and orthogonal matching pursuit (OMP) [13]. Both select the atoms for a certain data vector in a sequential and greedy manner. Other approaches are basis pursuit (BP) [27] and the focal under determined system solver (FOCUSS) [30], which replace the ℓ^0 norm with a ℓ^1 norm and a ℓ^p norm ($p \leq 1$), respectively. Also NMF with ℓ^1 sparsity constraints [25] can be counted to this group of methods.

The OMP algorithm is described in detail in Algorithm (1). In the initial steps 1-2, the residual $\mathbf{r}$ is defined as the data vector $\mathbf{x}$ and the index vector $\mathbf{z}$ is defined as zero dimensional vector. In each execution of the *for*-loop (steps 3-10), one atom and its corresponding coefficient is selected. In step 4, the scalar projection of the residual in the direction of every dictionary atom is calculated, where without loss of generality we can constrain the atoms to have unit length, i.e. $\|\mathbf{w}_k\| = 1,\ 1 \leq k \leq K$. In step 5, the index of the atom which approximates the residual best is determined, by searching the scalar resolute with the largest absolute value. In step 6, the index vector is replaced with the concatenation of the old index vector and the index selected in the previous step. In the steps 7-8, the data vector is projected into the space of the atoms collected so far, where $\mathbf{W}_z$ denotes the sub dictionary containing only the atoms indicated by $\mathbf{z}$, and $\mathbf{W}_z^+$ is its Moore-Penrose inverse. Finally in step 9, the new residual is defined as the difference between $\mathbf{x}$ and its reconstruction $\hat{\mathbf{x}}$.

Algorithm 1 Orthogonal Matching Pursuit (OMP)

```
 1: $\mathbf{r} \leftarrow \mathbf{x}$
 2: $\mathbf{z} = [\ ]$
 3: for l = 1:L do
 4:     $\mathbf{a} = \mathbf{W}^T \mathbf{r}$
 5:     $z^* = \arg\max |\mathbf{a}|$
 6:     $\mathbf{z} \leftarrow [\mathbf{z}, z^*]$
 7:     $\mathbf{c} = \mathbf{W}_{\mathbf{z}}^+ \mathbf{x}$
 8:     $\hat{\mathbf{x}} = \mathbf{W}_{\mathbf{z}} \mathbf{c}$
 9:     $\mathbf{r} \leftarrow \mathbf{x} - \hat{\mathbf{x}}$
10: end for
```

After the algorithm has terminated, $\mathbf{z}$ contains the indices of the selected atoms and $\mathbf{c}$ holds the corresponding coefficients, i.e. $h_{z_i} = c_i$. It is obvious that the residual $\mathbf{r}$ is reduced in every step, and that $\mathbf{r}$ is always orthogonal to $\mathbf{x}$. Also, for small L it is empirically verified that OMP delivers a close to optimal solution [13].

3.2 Enhancing the Dictionary

The dictionary is updated in sequential manner, i.e. the single atoms are enhanced in random sequence, while the rest of the dictionary is kept fixed. When $\mathbf{w}_k$ denotes the atom which shall be enhanced, then we can reformulate the objective Eq. (3.3) as

$$\begin{aligned} E_L &= \left\| \mathbf{X} - \sum_{j=1}^{K} \mathbf{w}_j \mathbf{h}^j \right\|_F^2 \\ &= \left\| \left(\mathbf{X} - \sum_{j \neq k}^{K} \mathbf{w}_j \mathbf{h}^j \right) - \mathbf{w}_k \mathbf{h}^k \right\|_F^2 \\ &= \left\| \mathbf{E}(k) - \mathbf{w}_k \mathbf{h}^k \right\|_F^2, \end{aligned} \tag{3.5}$$

where $\mathbf{h}^k$ denotes the k^{th} row of $\mathbf{H}$. Clearly, the error according to Eq. (3.5) would be minimal if $\mathbf{E}(k) = \mathbf{w}_k\mathbf{h}^k$. When the singular value decomposition (SVD) of $\mathbf{E}(k)$ is given as

$$\mathbf{E}(k) = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T, \tag{3.6}$$

then it is known that the product $\mathbf{u}_1\Sigma_{1,1}(\mathbf{v}_1)^T$ is the best rank 1 approximation of $\mathbf{E}(k)$. The best solution is therefore to set $\mathbf{w}_k = \mathbf{u}_1$ and $\mathbf{h}^k = \Sigma_{1,1}(\mathbf{v}_1)^T$. However, if we did this step immediately, we would destroy the sparse structure of the coefficient matrix, since maximal L elements in each column of $\mathbf{H}$ are nonzero as a result of the sparse coding step. The SVD approximation would introduce many additional nonzero elements in entries which held zeros before. The solution to this problem is to restrict the error matrix $\mathbf{E}(k)$ and the coefficient vector $\mathbf{h}^k$ to those columns where $\mathbf{h}^k$ is nonzero. Let ω_k be a list which contains the indices of the nonzero elements of $\mathbf{h}^k$, i.e.

$$\omega_k = \{i|1 \leq i \leq N, \mathbf{h}^k(i) \neq 0\}. \tag{3.7}$$

We can define a sub error matrix $\hat{\mathbf{E}}(k)$ with the columns of $\mathbf{E}(k)$ which are indexed by ω_k:

$$\hat{\mathbf{E}}(k) = \mathbf{E}(k)_{\omega_k}. \tag{3.8}$$

Using $\hat{\mathbf{E}}(k)$, we can define a modified objective:

$$\hat{E}_L = \left\|\hat{\mathbf{E}}(k) - \mathbf{w}_k\mathbf{h}^k_{\omega_k}\right\|_F^2. \tag{3.9}$$

The modified objective $\hat{E}_L$ is evaluated only for those training examples, where the atom $\mathbf{w}_k$ is used, i.e. where the corresponding coefficients are nonzero. Now we can use the SVD of $\hat{\mathbf{E}}(k)$ to replace the dictionary atom and the support of the coefficient vector. When the SVD of the restricted error matrix is given as

$$\hat{\mathbf{E}}(k) = \mathbf{U}\,\mathbf{\Sigma}\,\mathbf{V}^T, \tag{3.10}$$

we set $\mathbf{w}_k \leftarrow \mathbf{u}_1$ and $\mathbf{h}^k_{\omega_k} \leftarrow \Sigma_{1,1}\;(\mathbf{v}_1)^T$.

After each update, the dictionary is heuristically corrected by reinitializing atoms which are rarely used or which are too similar to other atoms. The M-step and the E-step are alternated for a fixed number of iterations, or until a convergence criterion is met. The K-SVD algorithm is summarized in Algorithm (2).

3.3 Efficient Version of K-SVD

Rubinstein developed in [31] a more efficient version of K-SVD. Every training signal has to be sparse coded with the current dictionary, which can be very time consuming. Rubinstein optimized the OMP algorithm (see Algorithm 1), by using a Cholesky factorization which allows to reuse the pseudo inverse of the last iteration. This enhanced algorithm called Batch-OMP performs superior compared to OMP, especially for large training sets.

However, the main effort in K-SVD comprises the SVD of large error matrices for each atom. Note, that it is not necessary to calculate the full SVD, but only the first singular value and its corresponding singular vectors. The original implementation by Aharon et

Algorithm 2 K-SVD

```
Initialize dictionary $\mathbf{W}$
for i = 1:numIter do
    Sparse code data $\mathbf{X}$, resulting in $\mathbf{H}$ (e.g. Algorithm (1))
    $\mathbf{r} = \text{randperm}(K)$
    for c = 1:K do
        $k = r_c$
        $\omega_k = \{i | 1 \leq i \leq N, \mathbf{h}^k(i) \neq 0\}$
        $\mathbf{E}(k) = \left(\mathbf{X} - \sum_{j \neq k}^{K} \mathbf{w}_j \, \mathbf{h}^j\right)$
        $\hat{\mathbf{E}}(k) = \mathbf{E}(k)_{\omega_k}$
        Calculate SVD: $\hat{\mathbf{E}}(k) = \mathbf{U} \, \mathbf{\Sigma} \, \mathbf{V}^T$
        $\mathbf{w}_k \leftarrow \mathbf{u}_1$
        $\mathbf{h}^k_{\omega_k} \leftarrow \Sigma_{1,1} \, (\mathbf{v}_1)^T$
    end for
    Heuristically correct dictionary
end for
```

al. [7] already took use of this fact by using the function `svds` in Matlab, which calculates only the first singular terms. However, Rubinstein noted in [31] that for an arbitrary matrix $\mathbf{E}$ the iterative process

$$\begin{aligned} \mathbf{w} &\leftarrow \mathbf{E} \, \mathbf{h} \\ \mathbf{w} &\leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|} \\ \mathbf{h} &\leftarrow \mathbf{E}^T \, \mathbf{w}. \end{aligned} \tag{3.11}$$

converges to the first singular terms of $\mathbf{E}$. In practice it can be observed that already the first iteration of this algorithm already comes close to the optimal solution. Therefore, the exact calculation of the first singular terms can be replaced with one iteration of Eq. (3.11). Since also K-SVD proceeds in iterations, the iterative SVD approximation will start with better and better initial values. The replacement of the exact SVD with one iteration of the approximation algorithm results in a large speed up while resulting in close to optimal results. Another advantage using the iterative SVD approximation is that an explicit calculation of the error matrix can be avoided, which is time and memory consuming. To see this fact, Eq. (3.11) is rewritten as:

$$\begin{aligned} \mathbf{E} &= \mathbf{X} - \mathbf{W} \, \mathbf{H} \\ \mathbf{w} &\leftarrow \mathbf{X} \, \mathbf{h} - \mathbf{W} \, \mathbf{H} \, \mathbf{h} \\ \mathbf{w} &\leftarrow \frac{\mathbf{w}}{\|\mathbf{w}\|} \\ \mathbf{h} &\leftarrow \mathbf{X}^T \, \mathbf{w} - \mathbf{W} \, \mathbf{H}^T \, \mathbf{w}. \end{aligned} \tag{3.12}$$

3.4 Nonnegative Version of K-SVD

When the data is known to be nonnegative, it is also plausible to constrain the dictionary and the coefficients to be nonnegative. Aharon et al. [12] modified K-SVD in order to introduce nonnegativity constraints. For the sparse coding stage, $\mathbf{H}$ is uniformly initialized with a small positive number and several iterations of the NMF update rule for $\mathbf{H}$ only are performed (see Eq. (2.15)). After that, the L dictionary atoms with the largest coefficients are selected for every data vector $\mathbf{x}_k, 1 \leq k \leq N$. Using these L atoms, $\mathbf{x}_k$ is approximated using the nonnegative LS solver described in [32].

As in the dictionary update of standard K-SVD, the atoms are are enhanced in a randomly picked sequence. Also, as in the original algorithm, the error matrix $\mathbf{E}(k)$ is restricted to the support of $\mathbf{h}^k$, i.e.

$$\begin{aligned} \omega_k &= \{i | 1 \leq i \leq N, \mathbf{h}^k(i) \neq 0\} \\ \hat{\mathbf{E}}(k) &= \mathbf{E}_{\omega_k}(k). \end{aligned}$$

However, the replacement of $\mathbf{w}_k$ and $\mathbf{h}^k_{\omega_k}$ with the first singular terms of the restricted error matrix would generally violate the nonnegativity constraints. Therefore, the SVD in Algorithm (2) (step 10) is replaced by an iterative method, similar to the iterative SVD approximation (see Eq. (3.11)), with an additional projection to the nonnegative space after each iteration. This operation is described in Algorithm (3), where $[\cdot]_+$ denotes the element wise maximum of the argument and zero. As in standard K-SVD, any arbitrary nonnegative sparse coder can be used for the sparse coding stage. In the next section we introduce an alternative sparse coding method, a modified version of OMP which implements nonnegativity constraints.

Algorithm 3 Nonnegative SVD approximation

```
Calculate SVD: $\hat{\mathbf{E}}(k) = \mathbf{U}\ \mathbf{\Sigma}\ \mathbf{V}^T$
$\mathbf{w} = [\mathbf{U}_1]_+$
$\mathbf{h} = [\mathbf{V}_1]_+$
for i = 1:numIter do
    $\mathbf{w} \leftarrow \left[\frac{\hat{\mathbf{E}}(k)\,\mathbf{h}}{\mathbf{h}^t\,\mathbf{h}}\right]_+$
    $\mathbf{h} \leftarrow \left[\frac{\mathbf{w}^T\,\hat{\mathbf{E}}(k)}{\mathbf{w}^T\,\mathbf{w}}\right]_+$
end for
$a = \mathbf{w}^T\,\mathbf{w}$
$\mathbf{w} = \frac{\mathbf{w}}{a}$
$\mathbf{h} = a\,\mathbf{h}$
```

3.5 Nonnegative Matching Pursuit

When we inspect the OMP algorithm (Algorithm (1)), we see that the first point where we can violate nonnegativity is step 5:

$$z^* = \arg\max|\mathbf{a}|. \tag{3.13}$$

The vector $\mathbf{a}$ holds the scalar resolutes of the residual when projected into the direction of the atoms. Since we are searching for the atom with the maximal *absolute* scalar resolutes, we also allow atoms with negative scalar resolute. If we remove the absolute operator in

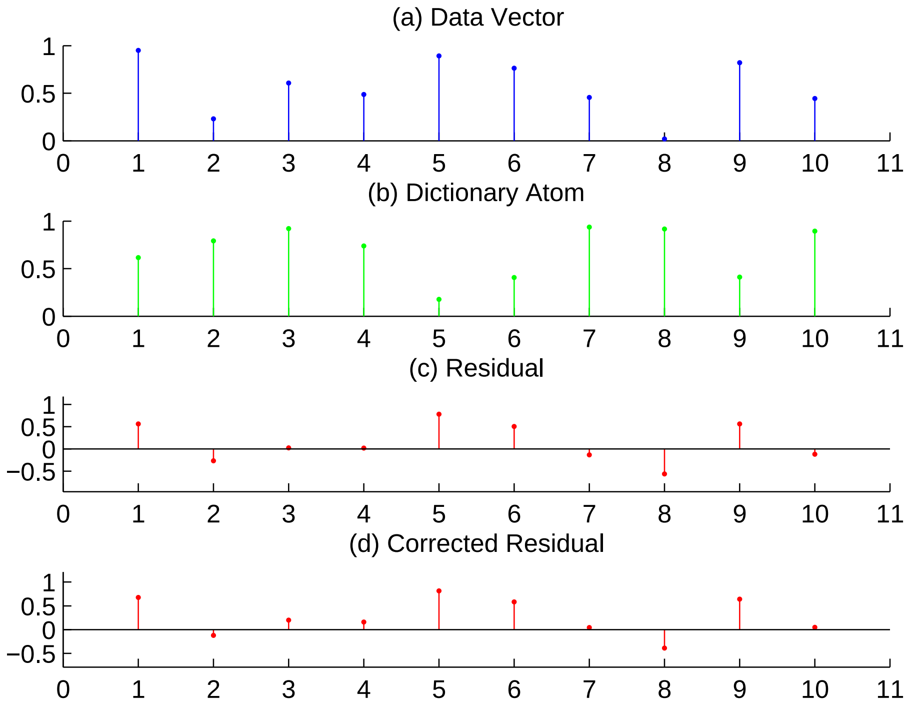


Figure 3.1: Least squares solution results into many negative residual values.

Eq. (3.13), then we are searching for the actual maximum of the projections, which is most probably positive, i.e. we change step 5 of OMP into

$$z^* = \arg\max \mathbf{a}. \tag{3.14}$$

In the case that the largest scalar resolutes is also negative, the algorithm has to terminate.

The second point were OMP can violate the nonnegativity constraint is step 7, were we project the data vector into the subspace of the collected atoms:

$$\mathbf{c} = \mathbf{W}_{\mathbf{z}}^{+}\,\mathbf{x}. \tag{3.15}$$

In general, the multiplication of the data vector with the pseudo inverse will also return negative values. The remedy to this problem is to replace Eq. (3.15) with a nonnegative LS solution. We can use the already mentioned algorithm in [32] or several iterations of the NMF update rule for $\mathbf{H}$ only, where $\mathbf{x}$ corresponds to $\mathbf{X}$, $\mathbf{c}$ to $\mathbf{H}$, and $\mathbf{W}_{\mathbf{z}}$ to $\mathbf{W}$ in Eq. (2.15). In practice we observed that even one iteration of NMF produces good results.

Although these modifications already guarantee nonnegativity, we still have to deal with a problem which is illustrated in Figure (3.1). When we approximate the data vector (a) with the atom (b) in least squares sense, the residual (c) will generally have many negative entries. Since we always reduce the residual, we will quickly arrive at a point where most of the residuals entries are negative, which can not be approximated with nonnegative atoms any more. This means that the algorithm has to terminate too early, when no positive z^* in Eq. (3.14) can be found. We can alleviate this problem by scaling the least squares solution with a correction factor $0 < f < 1$. In Figure (3.1) (d) we see a version of the residual where the optimal coefficient was scaled with 0.7. We see that the residual has less negative values and more energy in positive entries. Instead of scaling all entries of $\mathbf{c}$, we scale only the recently selected atoms with heuristic correction factors of 0.7, 0.8 and 0.9 for the last three collected atoms. A detailed description of the modified version of OMP, which we call nonnegative matching pursuit (NMP), is given in Algorithm (4). The scaling with the heuristic correction factors is realized via the multiplication with a diagonal matrix Δ. The function $\mathrm{diag}(\mathbf{a})$ returns a diagonal matrix with the elements of the vector $\mathbf{a}$ written in the main diagonal.

Algorithm 4 Nonnegative Matching Pursuit

```
 1: $\mathbf{f} = [0.9, 0.8, 0.7]$
 2: $\mathbf{r} \leftarrow \mathbf{x}$
 3: $\mathbf{z} = [\ ]$
 4: $\mathbf{c} = [\ ]$
 5: for l = 1:L do
 6:     $\mathbf{a} = \mathbf{W}^T \mathbf{r}$
 7:     $z^* = \arg\max \mathbf{a}$
 8:     $a^* = \max \mathbf{a}$
 9:     if $a^* < 0$ then
10:         Terminate
11:     end if
12:     $\mathbf{z} \leftarrow [\mathbf{z}, z^*]$
13:     $\mathbf{c} \leftarrow [\mathbf{c}, a^*]$
14:     $\mathbf{c} \leftarrow \mathbf{c} \otimes \frac{(\mathbf{W}_\mathbf{z}^T \mathbf{x})}{(\mathbf{W}_\mathbf{z}^T \mathbf{W}\, \mathbf{c})}$
15:     if $l \leq 3$ then
16:         $\Delta = \mathrm{diag}(\mathbf{f}(3 - (l-1) : 3))$
17:     else
18:         $\mathbf{f} \leftarrow [1, \mathbf{f}]$
19:         $\Delta = \mathrm{diag}(\mathbf{f})$
20:     end if
21:     $\hat{\mathbf{x}} = \mathbf{W}_\mathbf{z}\ \Delta\ \mathbf{c}$
22:     $\mathbf{r} \leftarrow \mathbf{x} - \hat{\mathbf{x}}$
23: end for
```

A very similar algorithm called nonnegative orthogonal matching pursuit (NOMP) was proposed by Yang et al. [33], which was not known by us when we developed NMP. There are two differences between NMP and NOMP. Firstly, NOMP does not perform the heuristic correction of the coefficients, thus it can be called *orthogonal*, while in NMP the residual is on purpose not orthogonal. Secondly, in NOMP the estimation of the coefficients $\mathbf{c}_k$ (which corresponds to step 14 in Algorithm (4)) is achieved via the multiplication with the Moore-Penrose inverse, as in Eq. (3.15). However, this step will generally introduce negative values for some entries of $\mathbf{c}$, which means that the nonnegativity constraints can be violated.

3.6 Beam Search NMP

NMP and the original OMP algorithm are greedy algorithms, since they select the locally best atom without the possibility to change the selection in a later step. We can design an intermediate algorithm between the optimal and intractable solution, and the affordable but suboptimal matching pursuit. This algorithm, which we call beam search NMP (BS-NMP), is described in Algorithm (5). The set s contains the found solutions, where a solution is represented with a triplet $\langle \mathbf{r}, \mathbf{z}, \mathbf{c} \rangle$, i.e. the residual, the index vector and the corresponding coefficient vector. The solution set s is initialized with a single trivial solution, where the selection vector $\mathbf{z}$ and the coefficient vector $\mathbf{c}$ are empty and the residual is the data vector $\mathbf{x}$. In every iteration of l, we build up to M new solutions for every solution out of s. These new solutions are inserted into a second solution set s^*, where s^* replaces s at the end of the iteration.

Instead of selecting only the best atom in each iteration (step 7 in Algorithm (4)), we select the M best atoms, i.e. the M atoms with the largest scalar resolute. This operation is realized using a sorting algorithm in step 12. For each of these M atoms, we run steps 14-22 of Algorithm (4)), where different solutions for $\mathbf{z}$, $\mathbf{c}$ and $\mathbf{r}$ are calculated and inserted into s^*. To assure nonnegativity, we test for a nonnegative scalar resolute in step 14. Therefore it is possible that less than M new solutions are found.

After T iterations we start to prune the solution set to the M^T best solutions (steps 25-31), since the solution space grows exponentially.

Finally after L iterations, we take the best branch as result. Dependent on the parameters M and T we can control the size of the search space. For $M = 1$ and $T = 1$, the algorithm reduces to NMP, while for $M = K$ and $T = L$, we perform a redundant full search over all possible combinations.

3.7 NMF with ℓ^0 constraints

Although the dictionary update of nonnegative K-SVD (Algorithm (3)) successfully introduces nonnegativity constraints to K-SVD, this method nevertheless is a circumvention of the problem. Nonnegativity is artificially achieved by simply setting negative values to zero. Therefore, we propose to use the NMF update rules (see Eq. (2.15)) for the dictionary update. NMF is the natural choice for this step, since it reduces the objective $\|\mathbf{X} - \mathbf{W}\,\mathbf{H}\|_F^2$ while maintaining nonnegativity. Additionally, NMF does not destroy the sparse structure of $\mathbf{H}$, since the update rule consists of a Hadamard product of $\mathbf{H}$ itself with another matrix term. Therefore, if an element of $\mathbf{H}$ is zero before the update, it will also be zero afterwards. This algorithm does not use a SVD in the dictionary update step and should therefore be called NMF with ℓ^0 constraints (NMFℓ^0). The NMFℓ^0 algorithm is described in Algorithm (6).

We compared NMFℓ^0 and nonnegative K-SVD by applying both methods to the same synthetic data. A random nonnegative dictionary matrix $\mathbf{W}$ of size 500×100 was created (i.e. $K = 100,\ D = 500$). A 100×2500 coefficient matrix $\mathbf{H}$ was generated by setting 5 randomly selected entries per column to random values between 0 and 10, while setting all other entries to zero (i.e. $L = 5$). The synthetic data $\mathbf{X}$ is given as $\mathbf{X} = \mathbf{W}\,\mathbf{H}$, which results into $N = 2500$ training examples. Both NMFℓ^0 and nonnegative K-SVD were executed with the correct dictionary size $K = 100$ and sparsity factor $L = 5$. For both

Algorithm 5 Beam-Search NMP

```
$\mathbf{f} = [0.9, 0.8, 0.7]$
$s \leftarrow \emptyset$
$\mathbf{r} \leftarrow \mathbf{x}$
$\mathbf{z} = [\ ]$
$\mathbf{c} = [\ ]$
$s \leftarrow s \cup \langle \mathbf{r}, \mathbf{z}, \mathbf{c} \rangle$
for l = 1:L do
    $s^* \leftarrow \emptyset$
    for $\forall \tilde{s} \in s$ do
        $\langle \mathbf{r}, \mathbf{z}, \mathbf{c} \rangle \leftarrow \tilde{s}$
        $\mathbf{a} = \mathbf{W}^T \mathbf{r}$
        $[\mathbf{a}^*, \mathbf{z}^*] = \text{descendSort}(\mathbf{a})$
        for $m = 1 : M$ do
            if $\mathbf{a}^*(m) < 0$ then
                Break
            end if
            $\langle \mathbf{r}, \mathbf{z}, \mathbf{c} \rangle \leftarrow \tilde{s}$
            $\mathbf{z} \leftarrow [\mathbf{z}, \mathbf{z}^*(m)]$
            $\mathbf{c} \leftarrow [\mathbf{c}, \mathbf{a}^*(m)]$
            Perform steps 14-22 of Algorithm (4).
            $s^* \leftarrow s^* \cup \langle \mathbf{r}, \mathbf{z}, \mathbf{c} \rangle$
        end for
    end for
    $s \leftarrow s^*$
    if $l > T$ then
        for $\forall \tilde{s} \in s$ do
            $\langle \mathbf{r}, \mathbf{z}, \mathbf{c} \rangle \leftarrow \tilde{s}$
            $E(\tilde{s}) = \|\mathbf{r}\|^2$
        end for
        reduce $s$ to maximal $M^T$ elements with smallest $E(\tilde{s})$
    end if
end for
find $\tilde{s} \in s$ with smallest $E = \|\mathbf{r}\|^2$
$\langle \mathbf{r}, \mathbf{z}, \mathbf{c} \rangle \leftarrow \tilde{s}$
```

Algorithm 6 NMFℓ^0

```
Initialize dictionary $\mathbf{W}$
for i = 1:numIter do
    Sparse code data $\mathbf{X}$, resulting in $\mathbf{H}$ (e.g. Algorithm (4))
    for c = 1:numNMF do
        $\mathbf{H} \leftarrow \mathbf{H} \otimes \frac{(\mathbf{W}^T\mathbf{X})}{(\mathbf{W}^T\mathbf{W}\mathbf{H})}$
        $\mathbf{W} \leftarrow \mathbf{W} \otimes \frac{(\mathbf{X}\mathbf{H}^T)}{(\mathbf{W}\mathbf{H}\mathbf{H}^T)}$
    end for
end for
Heuristically correct dictionary
```

algorithms we used NMP in the sparse coding stage and 30 iterations in the M-step, i.e. 30 iterations of nonnegative SVD approximation (Algorithm (3)) in nonnegative K-SVD, and 30 NMF updates (Eq. (2.15)) in NMFℓ^0. Both methods were executed for 25 iterations, i.e. the E-step and the M-step were alternated 25 times. After each iteration, the root mean squared error (RMSE) was calculated:

$$\text{RMSE} = \sqrt{\frac{\|\mathbf{X} - \mathbf{W}\,\mathbf{H}\|_F^2}{(D\,N)}}. \tag{3.16}$$

This experiment was repeated 20 times and the RMSE was averaged over these 20 runs. Figure (3.2) shows the averaged RMSE as a function of the number of iterations. The standard deviation is represented with vertical bars. We see that NMFℓ^0 achieves a significantly lower reconstruction error than nonnegative K-SVD.

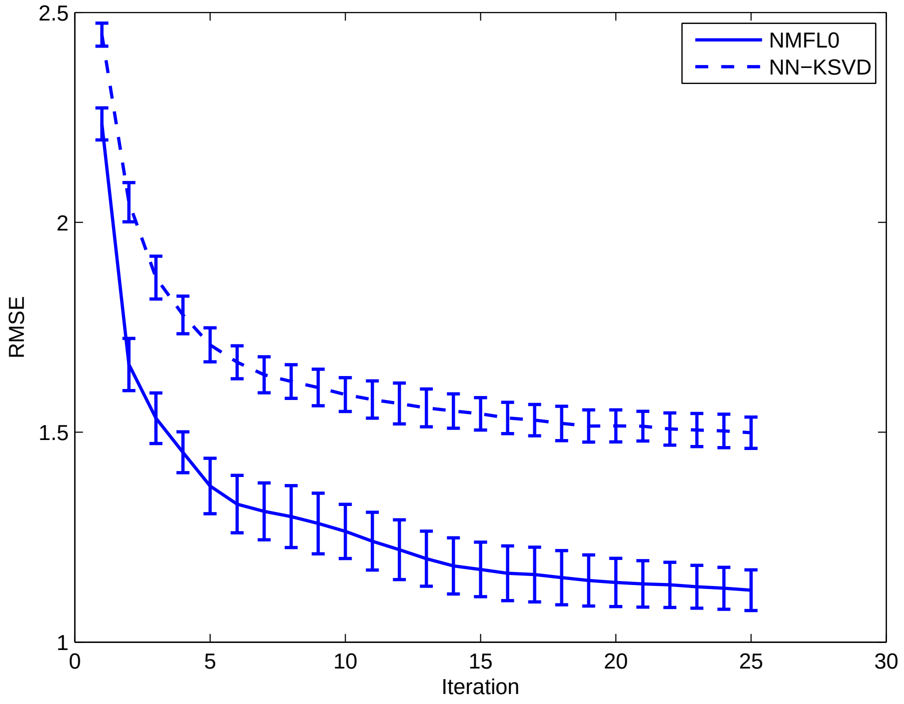


Figure 3.2: NMFℓ^0 (solid) compared to nonnegative K-SVD (dashed).

Chapter 4

Single Channel Source Separation

4.1 Simple Inference

Analog to the 2 stage approach using NMF (see Section (2.5)), we can apply K-SVD and NMFℓ^0 to the SCSS problem. This method, which we call simple inference, is illustrated in Figure (4.1). In the training stage we take for each of the M sources the magnitude spectrogram of source specific data. Further, we train a dictionary $\mathbf{W}^m$ for the m^{th} source, using either K-SVD or NMFℓ^0. For each source, we use the same parameters K and L.

In the separation stage, we concatenate the source specific dictionaries, i.e. $\mathbf{W} = \mathbf{W}^1 \cup \mathbf{W}^2 \cup \cdots \cup \mathbf{W}^M$. With the concatenated dictionary $\mathbf{W}$, we run an appropriate sparse coding technique, like OMP for K-SVD and NMP for NMFℓ^0, where maximal $M.L$ atoms are allowed to be used per spectrogram column. Finally, we split the returned coefficient matrix $\mathbf{H}$ according to the original dictionaries, resulting in coefficient matrices $\mathbf{H}^m$, for each source. An approximation $\hat{\mathbf{S}}^m$ of the m^{th} magnitude spectrogram $\mathbf{S}^m$ is given as $\hat{\mathbf{S}}^m = \mathbf{W}^m \, \mathbf{H}^m$. When K-SVD is applied, some entries of $\hat{\mathbf{X}}^m$ can contain negative values, since K-SVD is not constrained to be nonnegative. In this case, we simply set negative values to a small positive number. Using these approximations, we can calculate binary masks $\mathbf{BM}^m$ or continuous masks $\mathbf{CM}^m$ for each source, according to Eq. (2.11) and Eq. (2.12). The approximation of the m^{th} source signal in time domain is given as

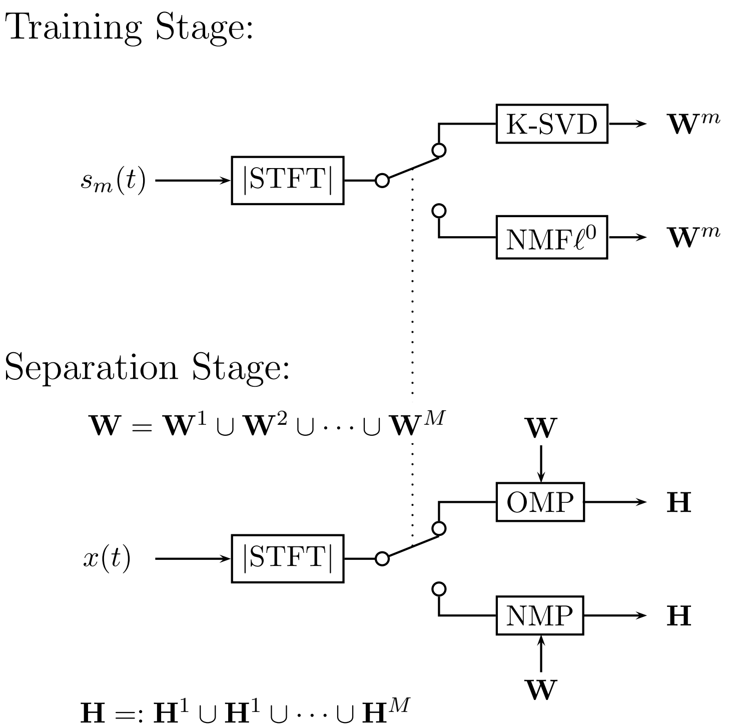
$$\hat{s}_m(t) = \text{ISTFT}(\mathbf{BM}^m \, \mathbf{X}), \tag{4.1}$$

in the case of a binary mask, and

$$\hat{s}_m(t) = \text{ISTFT}(\mathbf{CM}^m \, \mathbf{X}), \tag{4.2}$$

for the continuous mask. ISTFT denotes the inverse STFT, i.e. the column-wise inverse DFT and an overlap-add procedure (see also Section (2.4)).

We selected 2 female and 2 male speakers (which we call Female1, Female2, Male1 and Male2) from the speech data base by Cooke et al. [9]. We took approximately 3 minutes training speech for each speaker, where a sampling frequency of 16 kHz was used. For the calculation of the spectrograms we used a hamming window with 1024 samples length and 512 samples overlap. No zero padding was used and redundant frequency bins were discarded. For K-SVD and NMFℓ^0 we tried all combinations of the following parameter



$$\mathbf{W} = \mathbf{W}^1 \cup \mathbf{W}^2 \cup \cdots \cup \mathbf{W}^M$$

$$\mathbf{H} =: \mathbf{H}^1 \cup \mathbf{H}^1 \cup \cdots \cup \mathbf{H}^M$$

$$\hat{\mathbf{S}}^m = \mathbf{W}^m \, \mathbf{H}^m$$

$$\mathbf{BM}^m_{dn} = \begin{cases} 1 & \text{if } \hat{\mathbf{S}}^m_{dn} > \hat{\mathbf{S}}^l_{dn} \quad \forall l \neq m \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{CM}^m_{dn} = \frac{\hat{\mathbf{S}}^m_{dn}}{\sum_l \hat{\mathbf{S}}^l_{dn}}$$

BM: $\hat{s}_m(t) = \text{ISTFT}(\mathbf{BM}^m \, \mathbf{X})$

CM: $\hat{s}_m(t) = \text{ISTFT}(\mathbf{CM}^m \, \mathbf{X})$

Figure 4.1: Simple SCSS system for K-SVD and NMFℓ^0.

Female1	Female2	Male1	Male2
speaker 18	speaker 20	speaker 1	speaker 2
"lwixzs"	"lwwy2a"	"pbbp3s"	"lwwm2a"
"sbil4a"	"sbil2a"	"sbwo3a"	"sgai7p"
"prah4s"	"prbu5p"	"priv6p"	"priv3n"
"lbbc6a"	"lbbp1p"	"lbiq3a"	"lbbk3n"
"bgiz3p"	"bgwm5p"	"bgin3a"	"bgig8a"
"brae1n"	"brbe3n"	"brag1a"	"bgwb6a"
"lgix8a"	"lgwr2s"	"lrarzn"	"lgir7n"
"bbbk5p"	"bbbeza"	"bbbm3a"	"bbbmzs"
"prbo3p"	"prwb6s"	"pwaj6n"	"pwad2s"
"lwwd9n"	"lwwyzs"	"pbbv6n"	"lwws5p"

Table 4.1: Speaker labels, speaker ids used in [9] and test file names.

values of the dictionary size K and the maximal allowed number of atoms L:

$$\begin{aligned} K &= 50, 60, 70, 80, 90, 100, 120, 140, 160, \\ &\quad 180, 200, 250, 300, 350, 400, 450, 500 \\ L &= 1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40 \end{aligned}$$

In the separation stage we restricted ourselves to mixtures with two speakers only. We selected 10 utterances from each speaker as shown in Table (4.1), which were not used for training. Every speaker was combined with all other speakers, which results in 12 speaker pairs. For every speaker pair we mixed all possible utterances (i.e. 100 mixture utterances) with a signal to interference ratio (SIR) of 0 dB. The SIR after source separation is a measure for the achieved separation quality. In order to neglect phase distortion effects introduced by resynthesis, the SIR is calculated in the magnitude spectrogram domain, where $\mathbf{S}^m$ is the original source magnitude spectrogram and $\hat{\mathbf{S}}^m$ is its estimation:

$$\text{SIR} = 10 \log \left(\frac{\|\mathbf{S}^m\|_F^2}{\|\mathbf{S}^m - \hat{\mathbf{S}}^m\|_F^2} \right). \tag{4.3}$$

Figures (4.2) and (4.3) show the average SIR after demixing as a function of K, for K-SVD and NMFℓ^0, respectively. Each sub figure shows the SIR for a specific target speaker compared to the other three speakers as interfering speakers. Each interfering speaker is represented by a color and a marker. For every value of K, the mean over all mixture utterances and over all values of L was calculated. The solid lines show the result when a CM (see Eq. (2.12)) was used, while the dashed lines show the result for the BM (see Eq. (2.11)). The straight lines with an additional marker (black circle) show the result when an optimal mask (OM) is used. We can calculate the OM by using the original source spectrograms instead of approximations. The result with an OM gives an upper bound for the separation quality which can be achieved by refiltering. We see that the dictionary size does not seem to be important for this simple demixing system.

It is obvious that the CM is preferable to the BM, since the performance is always slightly better when the CM is used. Also for the other SCSS methods factorial max-VQ

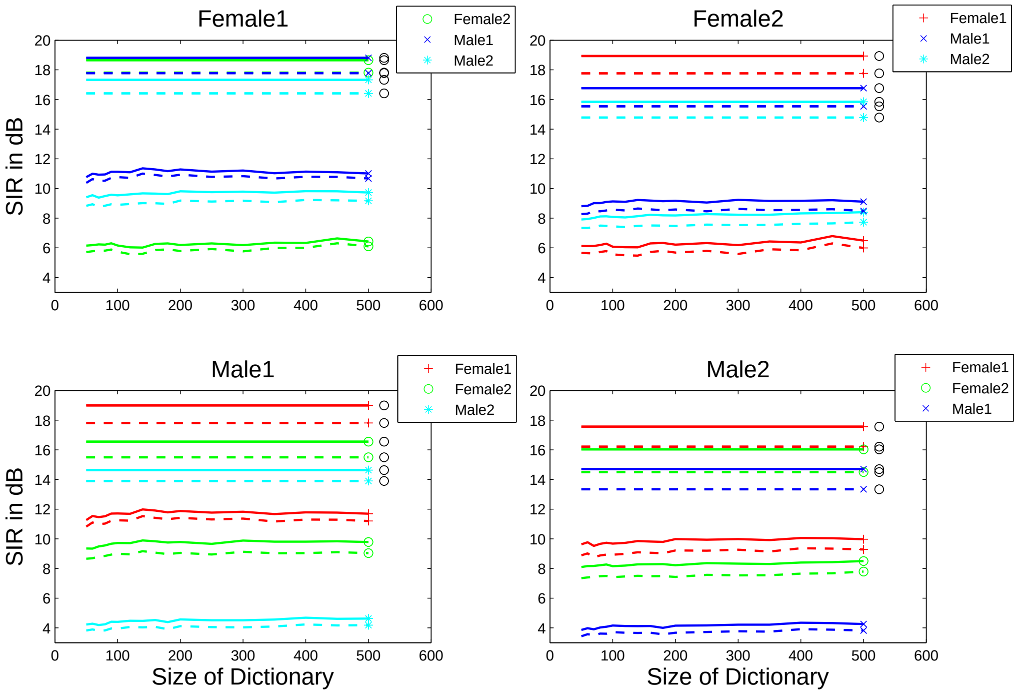


Figure 4.2: Mean SIR for K-SVD with BM (dashed) and CM (solid).

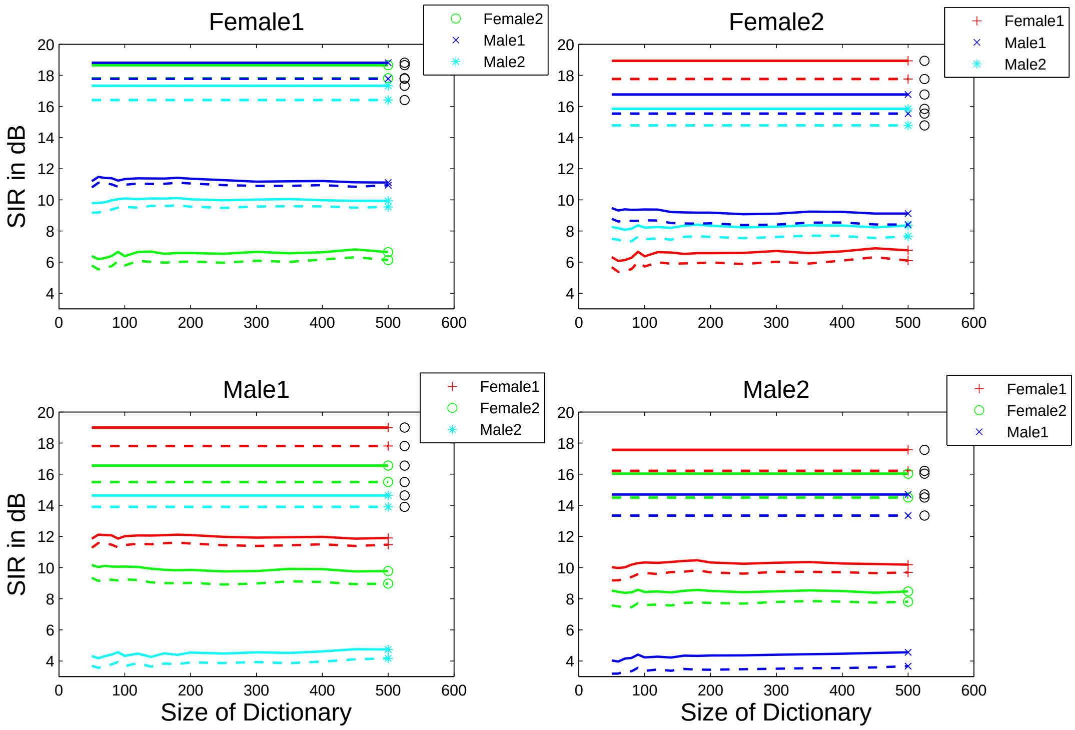


Figure 4.3: Mean SIR for NMFℓ^0 with BM (dashed) and CM (solid).

and NMF we observed a similar behavior. Therefore, we will only use the CM for the remainder of this thesis.

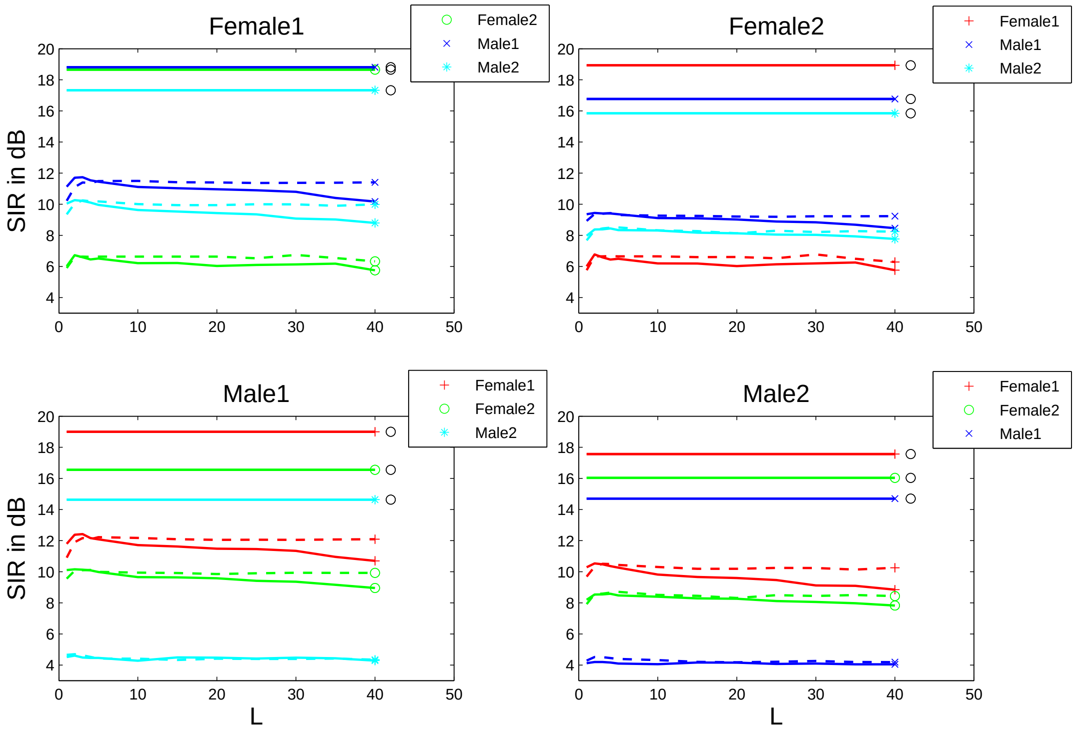


Figure 4.4: Mean SIR as function of L for K-SVD (solid) and NMFℓ^0 (dashed)

Figure (4.4) shows the average performance of K-SVD and NMFℓ^0 as a function of L, where the mean was taken over all mixture utterances and all values of K. For K-SVD, the SIR decreases with larger L, while the performance of NMFℓ^0 stays nearly constant. NMP (Algorithm (4)) terminates in step 10, when no atom with positive scalar resolute can be found. This means that NMP often selects less than the L allowed atoms, which introduces some robustness when this parameter is set too large. However, Figure (4.4) indicates that the optimal range for L is $2 \leq L \leq 5$.

Figure (4.5) compares the results of K-SVD, NMFℓ^0 and max-VQ, where we see that max-VQ performs best. This is understandable, since max-VQ uses the optimal solution and additional prior information, while K-SVD and NMFℓ^0 use a quick error minimization technique. On the other hand, the execution of K-SVD and NMFℓ^0 is much faster than max-VQ, while still giving considerable good results, especially for small K. When we compare K-SVD and NMFℓ^0, we see that their performance is approximately the same. This means that the nonnegativity constraints of NMFℓ^0 do not seem to have any advantage nor disadvantage for this simple SCSS system.

It is notable, that the SIR for both sparse coding techniques drop dramatically in comparison to max-VQ, when two speakers of the same gender have to be demixed. In this case, the spectra of the two speakers overlap to a larger extent than in the case of two speakers of different gender. Both K-SVD and NMFℓ^0 are generalizations of k-means, and therefore they provide a larger expressibility. However, this expressibility can easily

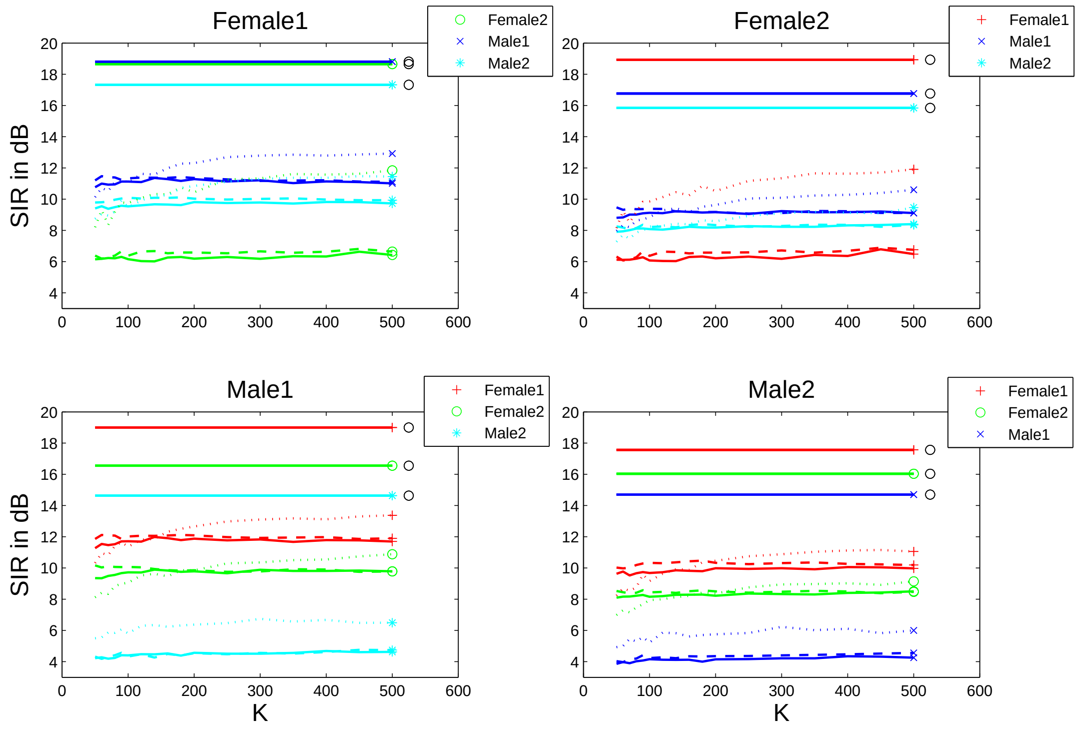


Figure 4.5: Mean SIR for K-SVD (solid), NMFℓ^0 (dashed) and max-VQ (dotted).

lead to a mismatch of the signal atoms, especially when the spectra of the two speaker are overlapping. We can compare this phenomenon to polynomial curve overfitting. When we use polynomials of higher and higher order (i.e. functions with higher expressibility), we can increasingly improve the interpolation quality for a given set of points, i.e. we can make the approximation error smaller and smaller. However, when this set of points is generated by some physical process, it is unlikely that a higher order polynomial represents this process correctly. For our task this means that although K-SVD and NMFℓ^0 are able to approximate the mixture data better than the factorial max-VQ model, this does not mean that the more sophisticated models automatically find the underlying causes, i.e. the real sources. According to this idea, we have to introduce additional constraints, in order to control the flexibility of K-SVD and NMFℓ^0.

However, we can try to improve the simple SCSS system with NMFℓ^0 by using BS-NMP in the sparse coding stage instead of NMP. In this case, the reconstruction error will be smaller, which should increase the separation quality. Figure (4.6) compares the results of the simple system using NMFℓ^0 with NMP (NMFℓ^0/NMP), NMFℓ^0 with BS-NMP (NMFℓ^0/BS-NMP) and max-VQ. Again, the SIR is plotted as a function of K, where $L = 5$ for both NMFℓ^0/NMP and NMFℓ^0/BS-NMP. For BS-NMP we selected the parameters $N = 4$ and $T = 2$ (see Section (3.6)). We see that the separation quality increases when BS-NMP is used due to a better error reduction. However, max-VQ still performs better in many cases, especially for large K. Furthermore, we still can observe a dramatical performance drop-off in the same gender case. This experiment confirms that error reduction alone does not lead to a successful identification of the sources.

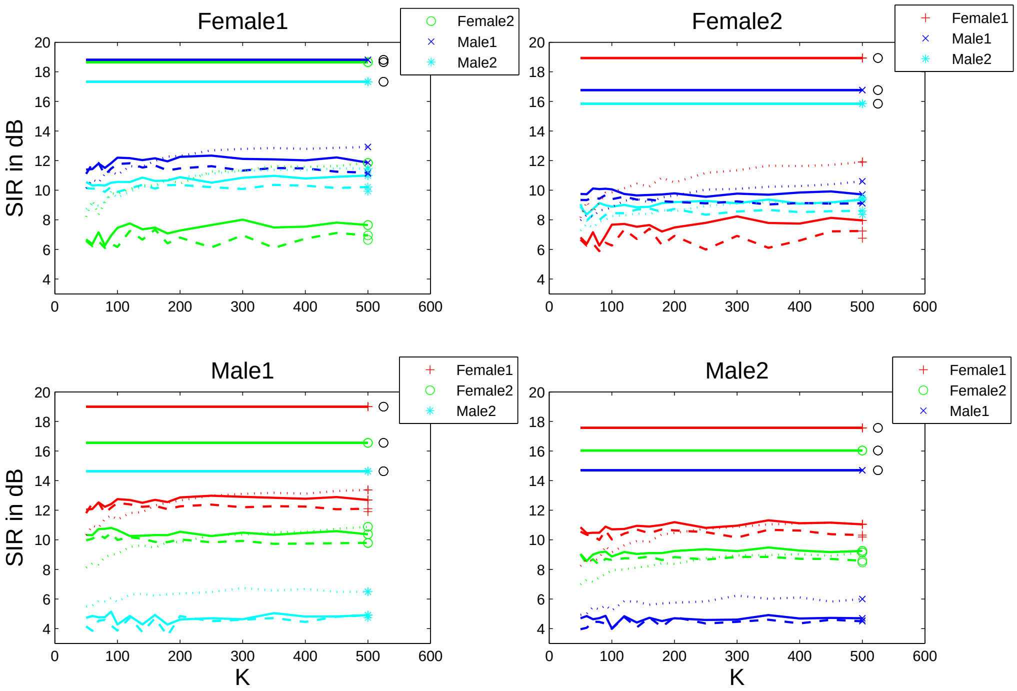


Figure 4.6: Mean SIR for NMFℓ^0/BS-NMP (solid), NMFℓ^0/NMP (dashed) and max-VQ (dotted).

4.2 Best Case Analysis

The results of the simple SCSS system in the last section are not satisfying. We would expect that K-SVD and NMFℓ^0 perform better, since they are generalized versions of k-means, the training algorithm of the max-VQ system. However, reasons for this low performance are that the simple system does not include any additional constraints nor prior information, and that quick but greedy and suboptimal error minimization techniques are applied. In this section we want to demonstrate, that the generalized versions of k-means *are* at least able to perform better. Also we want to determine, if the nonnegativity constraints of NMFℓ^0 have any advantage for the SCSS problem. Intuitively we would expect so, since we are dealing with magnitude spectrograms, i.e. nonnegative data.

Again, we executed the separation stage of the simple system, but this time we applied the sparse coders (OMP or respectively NMP) to the *original* source magnitude spectrograms, using the dictionary of the corresponding source. This means, we merely reconstructed the original spectrograms using a sparse coder. In practice of course this approach is useless, since if we had the original source spectrograms, we would not have to perform source separation. However, we performed this step merely to determine the best performance which *can* be achieved with K-SVD and NMFℓ^0. Although OMP and NMP do not return the optimal solution, let us assume that the result gives an approximate upper performance bound. Also for max-VQ we performed a best case analysis, by replacing each source spectrogram column with the closest code book entry.

Figure (4.7) shows the achieved performance for the optimal case. We see that the SIR

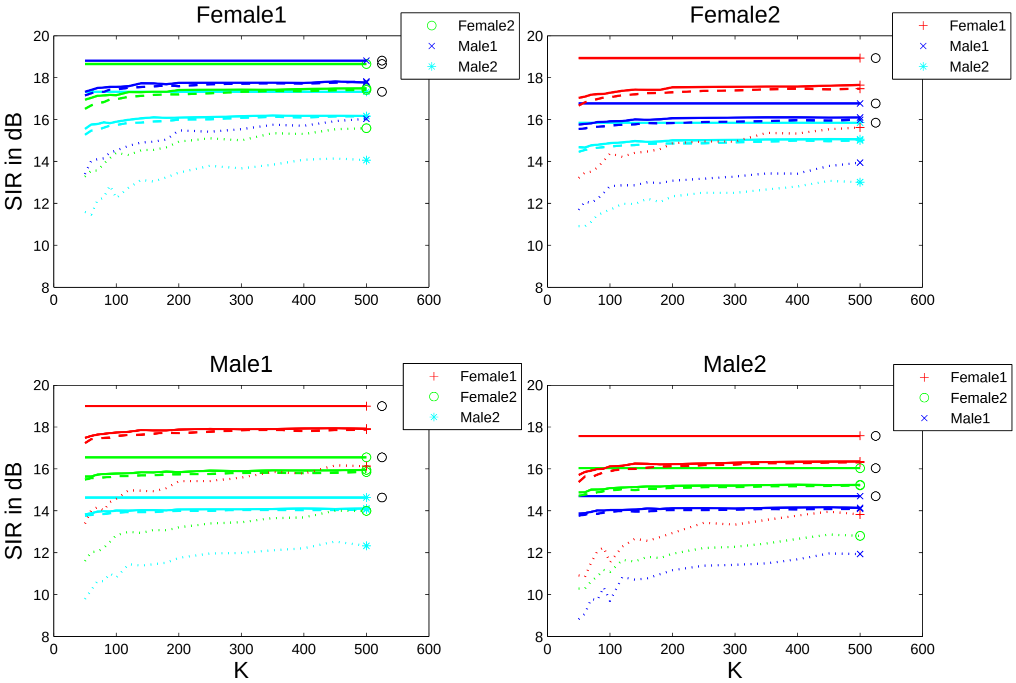


Figure 4.7: Result of Optimal Analysis for K-SVD (solid), NMFℓ^0 (dashed) and max-VQ (dotted).

for K-SVD and NMFℓ^0 is already close to the absolute maximum which can be achieved by using an OM. Also for $K = 500$, the SIR for both methods is approximately 2 dB higher than for max-VQ. This means, as expected, that both methods can perform better than max-VQ.

K-SVD always performs slightly better than NMFℓ^0, because the separation quality in the optimal case is reciprocal to the reconstruction error of the method. K-SVD can approximate the data better, since it is not constraint to be nonnegative as NMFℓ^0, which results in a better performance in the optimal case.

However, in the optimal analysis we were provided with a close to optimal $\mathbf{H}$, i.e. with close to optimal atom indices and corresponding coefficients. We can ask how our simple system would perform if we were provided with the optimal indices only, without coefficients. Therefore, we again performed the optimal analysis, but this time we discarded the values of the coefficients returned by OMP and NMP. Instead we found new coefficients by calculating an LS approximation of the mixture data, using the atoms of all sources indicated by the sparse coder. For K-SVD this task is solved analog as in Algorithm (1), steps 7 and 8, and for NMFℓ^0 we used the nonnegative LS solver [32].

Figure (4.8) shows the result of this experiment, compared to the optimal solution of max-VQ. NMFℓ^0 clearly achieves the highest separation quality. Also we can see that the performance of NMFℓ^0 is still quite close to the optimal solution with the OM. We can interpret this result as follows: When we managed to determine the optimal atom indices for NMFℓ^0 dictionaries, given the mixture data, then we can also infer suitable coefficients by using a simple nonnegative LS approximation. This means that we merely

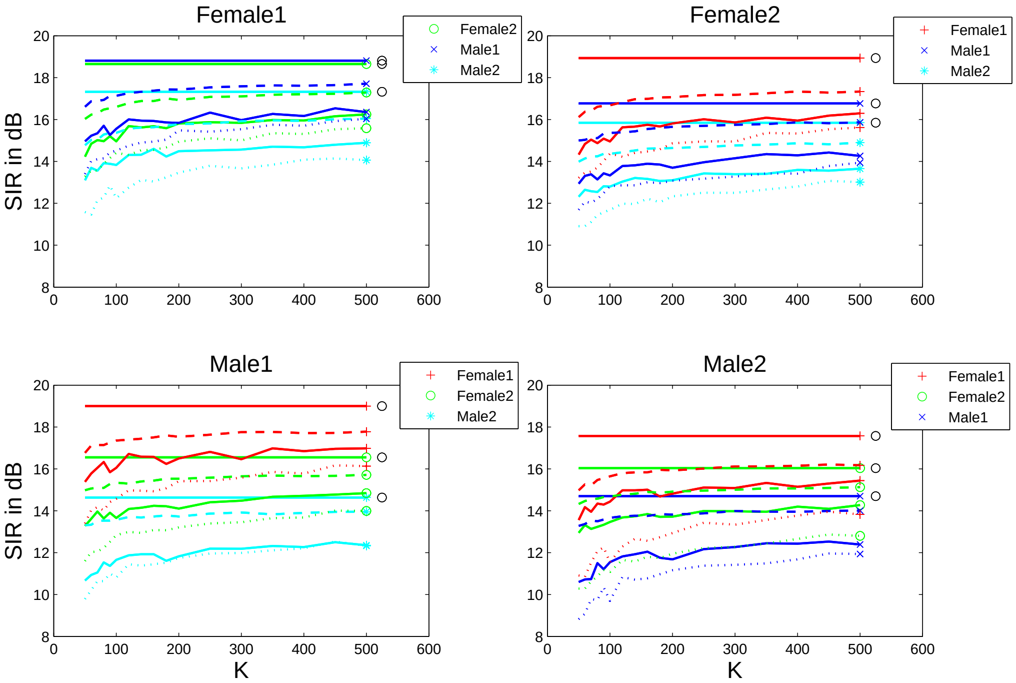


Figure 4.8: Result of LS solution with optimal selected atoms for K-SVD (solid) and NMFℓ^0 (dashed), and optimal max-VQ (dotted).

need to know a good selection of NMFℓ^0 atoms to achieve good separation quality. On the other hand, for K-SVD dictionaries, we would also need to infer suitable coefficient values, e.g. with gradient methods, to obtain similar or better performance. Therefore we can reduce the separation algorithm using NMFℓ^0 dictionaries to a search problem of appropriate atom indices.

4.3 Factorial Sparse Coder Model

In this section we propose a probabilistic factorial sparse coder (SC) model, analog to the factorial max-VQ system by Roweis [6]. The overall system for M interfering sources is shown in Figure (4.9). The mixture magnitude spectrogram column $\mathbf{x}$ is assumed to be the sum of the magnitude spectrogram columns of the sources, i.e.

$$\mathbf{x} = \sum_{m=1}^{M} \mathbf{s}^m. \tag{4.4}$$

The source spectrogram column $\mathbf{s}^m$ is modeled as the output $\hat{\mathbf{s}}^m$ of the m^{th} sparse coder using the dictionary $\mathbf{W}^m$, plus an additive noise term $\mathbf{n}^m$, for $1 \leq m \leq M$. The output of the m^{th} sparse coder is given as

$$\hat{\mathbf{s}}^m = \sum_{k=1}^{L} h^m_{z^m_k} \mathbf{w}^m_{z^m_k} = \mathbf{W}^m_{\mathbf{z}^m}\, \mathbf{h}^m_{\mathbf{z}^m}\ ,\ 1 \leq m \leq M, \tag{4.5}$$

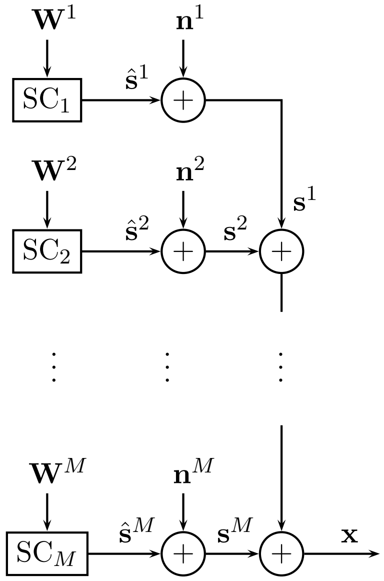


Figure 4.9: Factorial Sparse Coder Model.

where $\mathbf{W}^m_{\mathbf{z}^m}$ is the sub-dictionary of $\mathbf{W}^m$ and $\mathbf{h}^m_{\mathbf{z}^m}$ is the sub-vector of $\mathbf{h}^m$, with the atoms and the entries indicated by $\mathbf{z}^m$, respectively. As argued in the previous section, we assume that the coefficients are given by the nonnegative LS approximation of $\mathbf{x}$, using the atoms indicated by $\mathbf{z}^m$, i.e.

$$\begin{pmatrix} \mathbf{h}^1_{\mathbf{z}^1} \\ \mathbf{h}^2_{\mathbf{z}^2} \\ \vdots \\ \mathbf{h}^M_{\mathbf{z}^M} \end{pmatrix} = \arg\min_{\mathbf{h}} \left\| \mathbf{x} - \left(\mathbf{W}^1_{\mathbf{z}^1}\ \mathbf{W}^2_{\mathbf{z}^2} \ldots \mathbf{W}^M_{\mathbf{z}^M} \right) \mathbf{h} \right\|^2 \ , \text{ s.t. } h_k \geq 0,\ \forall k, \tag{4.6}$$

where the left hand side of Eq. (4.6) is the concatenation of $\mathbf{h}^m_{\mathbf{z}^m}$, $1 \leq m \leq M$, and $\left(\mathbf{W}^1_{\mathbf{z}^1}\ \mathbf{W}^2_{\mathbf{z}^2} \ldots \mathbf{W}^M_{\mathbf{z}^M} \right)$ denotes the the concatenated sub-dictionaries. We define $\hat{\mathbf{x}}$ as the LS approximation for given selections $\mathbf{z}^m$, $1 \leq m \leq M$, i.e.

$$\hat{\mathbf{x}} = \left(\mathbf{W}^1_{\mathbf{z}^1}\ \mathbf{W}^2_{\mathbf{z}^2} \ldots \mathbf{W}^M_{\mathbf{z}^M} \right) \begin{pmatrix} \mathbf{h}^1_{\mathbf{z}^1} \\ \mathbf{h}^2_{\mathbf{z}^2} \\ \vdots \\ \mathbf{h}^M_{\mathbf{z}^M} \end{pmatrix} = \sum_{m=1}^{M} \hat{\mathbf{s}}^m. \tag{4.7}$$

In the training stage of NMFℓ^0 we observed that the entries of the residual (i.e. the rows of the error matrix $\mathbf{E} = \mathbf{X} - \mathbf{W}\,\mathbf{H}$) are distributed similar to Laplacian distributions. The Laplacian distribution is given as

$$p_L(x|\mu, \lambda) = \frac{1}{2\,\lambda} e^{-\frac{|x-\mu|}{\lambda}}, \tag{4.8}$$

where μ is the mean value and λ is a form factor, which is given as a function of the variance of the distribution:

$$\lambda = \sqrt{\frac{E\{p(x)^2\}}{2}}. \tag{4.9}$$

Figure (4.10) shows the normalized histograms of 4 randomly selected entries of the NMFℓ^0 residual, where the data matrix $\mathbf{X}$ is the magnitude spectrogram of 3 minutes of speech from the data base by Cooke et al. [9]. Additionally, the plot shows fitted Laplacian distributions, with form factors calculated according to Eq. (4.9). The residual entries contain some large outliers, which would lead to a misestimation of the variance. Therefore we used only 90% of the residual samples which are closest to zero, in order to discard outliers and to gain a robust variance estimation. We see that the fitted Laplacian distributions match well with the normalized histograms.

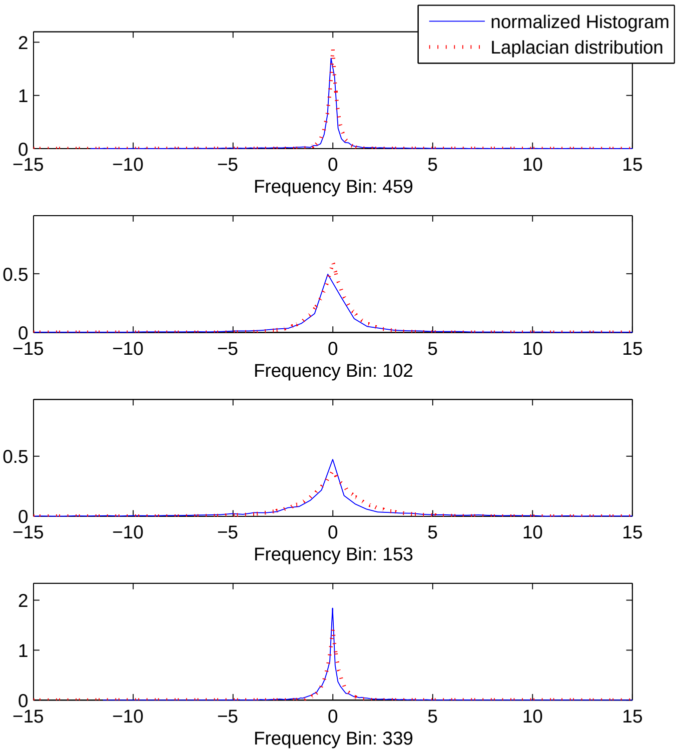


Figure 4.10: Normalized histograms of 4 randomly selected frequency bins and fitted Laplacian distributions.

Therefore we assume that the noise $\mathbf{n}^m$ $(1 \leq m \leq M)$ is distributed according to a Laplacian distribution with zero mean, and furthermore, that each frequency bin is independent from all other bins. Using Eq. (4.9) we can calculate the form factors $\boldsymbol{\lambda}^m = (\lambda_1^m, \lambda_2^m, \ldots, \lambda_D^m)$ for the noise vectors $\mathbf{n}^m$, $1 \leq m \leq M$, where the robust variance estimation is used. Furthermore, since $\mathbf{s}^m = \hat{\mathbf{s}}^m + \mathbf{n}^m$, we see that the source spectrogram column $\mathbf{s}^m$ is distributed according to a Laplace distribution with $\hat{\mathbf{s}}^m$ mean and form factors $\boldsymbol{\lambda}^m$, where all frequency bins are mutual independent:

$$p(s_d^m) = p_L(s_d^m | \hat{s}_d^m, \lambda_d^m), \; 1 \leq d \leq D, \; 1 \leq m \leq M. \tag{4.10}$$

Recalling Eq. (4.4), the mixture $\mathbf{x}$ is modeled as $\mathbf{x} = \sum_{m=1}^{M} \mathbf{s}^m$. The pdf of the sum of several independent random variables is the convolution of the individual pdfs. Therefore, the d^{th} frequency bin of $\mathbf{x}$ is distributed according to the convolution of M Laplacian distributions:

$$p(x_d) = p_L(s_d^1|\hat{s}_d^1, \lambda_d^1) * p_L(s_d^2|\hat{s}_d^2, \lambda_d^2) * \ldots * p_L(s_d^M|\hat{s}_d^M, \lambda_d^M) \tag{4.11}$$

For simplicity we restrict ourselves to the case $M = 2$. In this case, x_d is distributed according to

$$p(x_d|\hat{x}_d, \lambda_d^1, \lambda_d^2) = \frac{1}{2}\left[\frac{\lambda_d^1}{(\lambda_d^1)^2 - (\lambda_d^2)^2}\, e^{\frac{-|x_d - \hat{x}_d|}{\lambda_d^1}} + \frac{\lambda_d^2}{(\lambda_d^2)^2 - (\lambda_d^1)^2}\, e^{\frac{-|x_d - \hat{x}_d|}{\lambda_d^2}}\right], \tag{4.12}$$

where $\hat{s}_d^1 + \hat{s}_d^2$ was substituted by $\hat{x}_d$ (see Eq. (4.7)). For a derivation of Eq. (4.12) see Section (A.1). Using Eq. (4.12), we can define the likelihood of the selections $\mathbf{z}^1$ and $\mathbf{z}^2$:

$$p(\mathbf{x}|\mathbf{z}^1, \mathbf{z}^2) = p(\mathbf{x}|\hat{\mathbf{x}}(\mathbf{z}^1, \mathbf{z}^2), \boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2) = \prod_{d=1}^{D} p(x_d|\hat{x}_d, \lambda_d^1, \lambda_d^2). \tag{4.13}$$

The probability of a selection $\mathbf{z}^m$ is the joint probability of its entries, which is approximated by a Markov chain:

$$p(\mathbf{z}^m) = p(z_1^m, z_2^m, \ldots z_L^m) \approx p(z_1^m) \prod_{k=2}^{L} p(z_k^m|z_{k-1}^m), \; m \in \{1, 2\}. \tag{4.14}$$

The probabilities $p(z_1^m)$ and $p(z_k^m|z_{k-1}^m)$ can be estimated from the coefficient matrix in the training stage. According to Bayes theorem, the posterior probability of the selections $\mathbf{z}^1$ and $\mathbf{z}^2$, given the mixture data $\mathbf{x}$, is can be written as

$$p(\mathbf{z}^1, \mathbf{z}^2|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{z}^1, \mathbf{z}^2)p(\mathbf{z}^1, \mathbf{z}^2)}{p(\mathbf{x})}. \tag{4.15}$$

Assuming independent sources and using Eq. (4.13) and Eq. (4.14), we find

$$p(\mathbf{z}^1, \mathbf{z}^2|\mathbf{x}) \propto p(\mathbf{z}^1)p(\mathbf{z}^2) \prod_{d=1}^{D} p(x_d|\hat{x}_d, \lambda_d^1, \lambda_d^2). \tag{4.16}$$

The source separation problem is now defined as maximization of Eq. (4.16) with respect to $\mathbf{z}^1$ and $\mathbf{z}^2$. The approximations of the source spectrograms are found implicitly by the LS approximation of the data (Eq. (4.6)) and are given as $\hat{\mathbf{s}}^m = \mathbf{W}_{\mathbf{z}^m}^m \mathbf{h}_{\mathbf{z}^m}^m, \; m \in \{1, 2\}$.

Finding the optimal solution of Eq. (4.16) is an intractable problem, since we had to consider $\binom{L}{K}^2$ combinations. Therefore, we restrict the search space in order to find $\mathbf{z}^1$ and $\mathbf{z}^2$ with a high posterior. For this task, we use BS-NMP (Algorithm 5) with a changed pruning criterion: Instead of keeping the branches with the lowest approximation error in step 24, we prune the coding tree to the M^T branches with highest *a posterior* according to Eq. (4.16). In step 28, we select the branch with the highest posterior probability as final result.

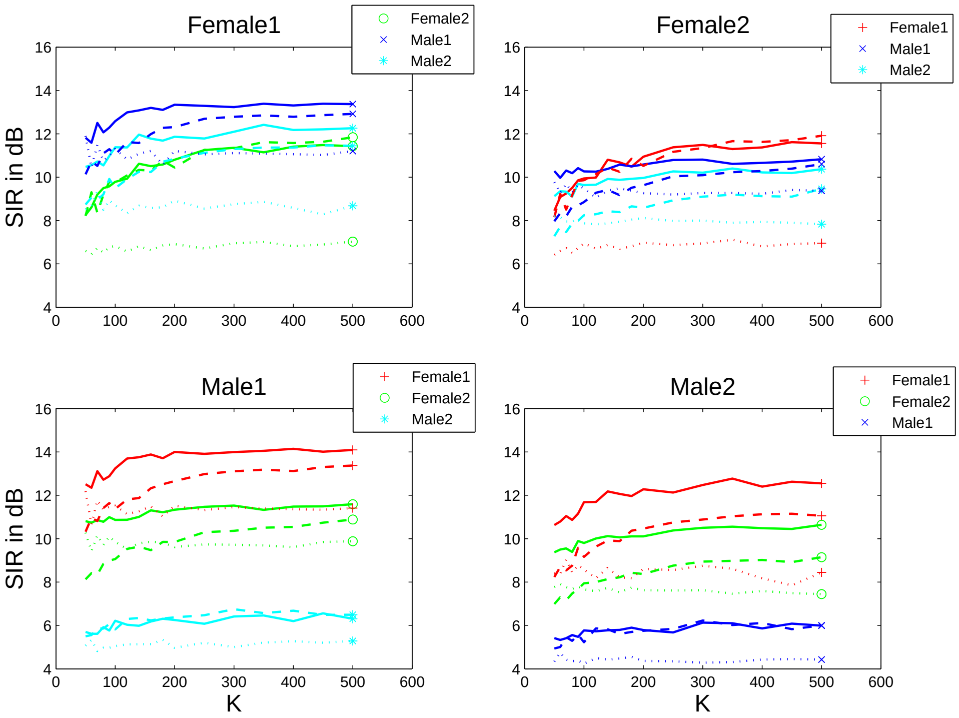


Figure 4.11: Mean SIR of factorial SC (solid), factorial max-VQ (dashed) and NMF (dotted).

Again we performed the separation experiment described in Section (4.1), where we used NMFℓ^0 dictionaries trained with $L = 4$, and set $M = 4$, $T = 2$ for BS-NMP with changed pruning criterion. Figure (4.11) compares the SIR of factorial SC, factorial max-VQ and NMF (see Section 2.5). We see that the factorial SC system clearly performs best. Only in two cases (Female1-Female2 and Female2-Female1) the max-VQ system performs slightly better. NMF has a significantly lower separation quality than NMFℓ^0 and in most cases it is also outperformed by factorial VQ. Only in some cases and $K < 100$, NMF achieves a better performance than factorial VQ. Figure (4.12) shows the standard deviation (in linear domain, not in dB) of the achieved SIR for all three systems. We see that the standard deviation is correlated with the achieved mean SIR, so that the results presented in Figure (4.11) seem to be little significant.

However, Figure (4.13) shows the percentage of examples where factorial SC, factorial max-VQ and NMF performed best, respectively. We see that in about 60 % of the examples factorial SC performed best, factorial max-VQ achieved in approximately 30 % the best performance and in less than 10 % (for $K > 100$) NMF had the best separation result. The percentage is calculated using all 1200 separation results per value of K (4 speakers, 3 interfering speakers, 100 mixed utterances: $4 \times 3 \times 100 = 1200$ experiments). Figure (4.14) shows the same percentage, where the 3 separation systems are compared pairwise. Both factorial SC and factorial max-VQ achieved in more than 75 % (for $K > 150$) a better result than NMF. We can also see that in more than 60 % factorial SC performed better than factorial max-VQ, for each value of K.

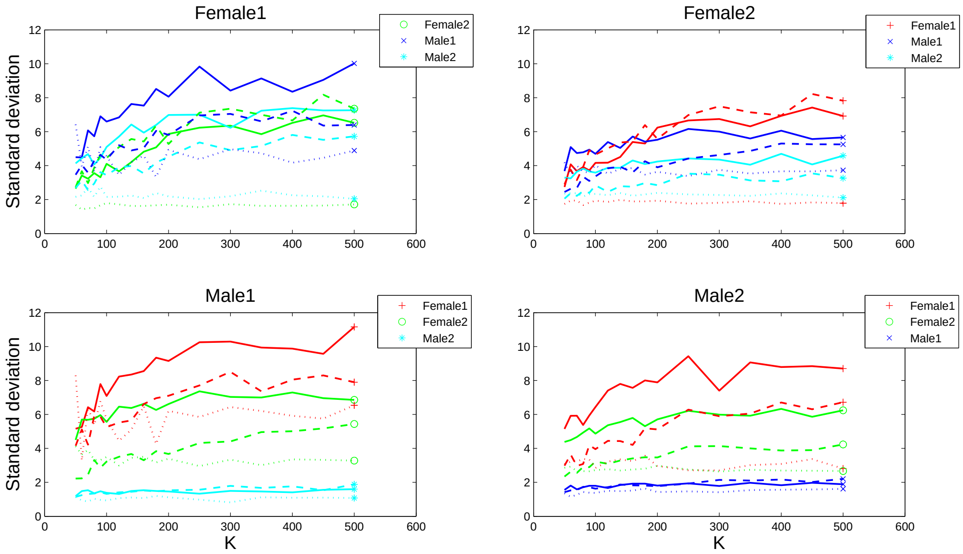


Figure 4.12: Standard deviation of the linear SIR for factorial SC (solid), factorial max-VQ (dashed) and NMF (dotted).

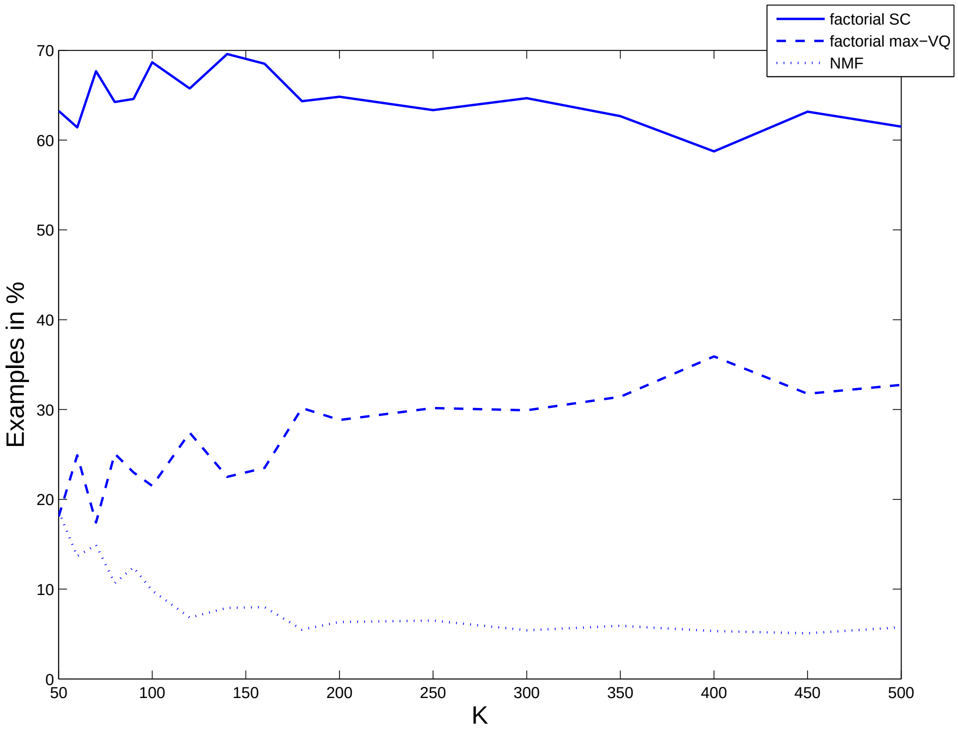


Figure 4.13: Percentage of examples where factorial SC (solid), factorial max-VQ (dashed) and NMF (dotted) performed best.

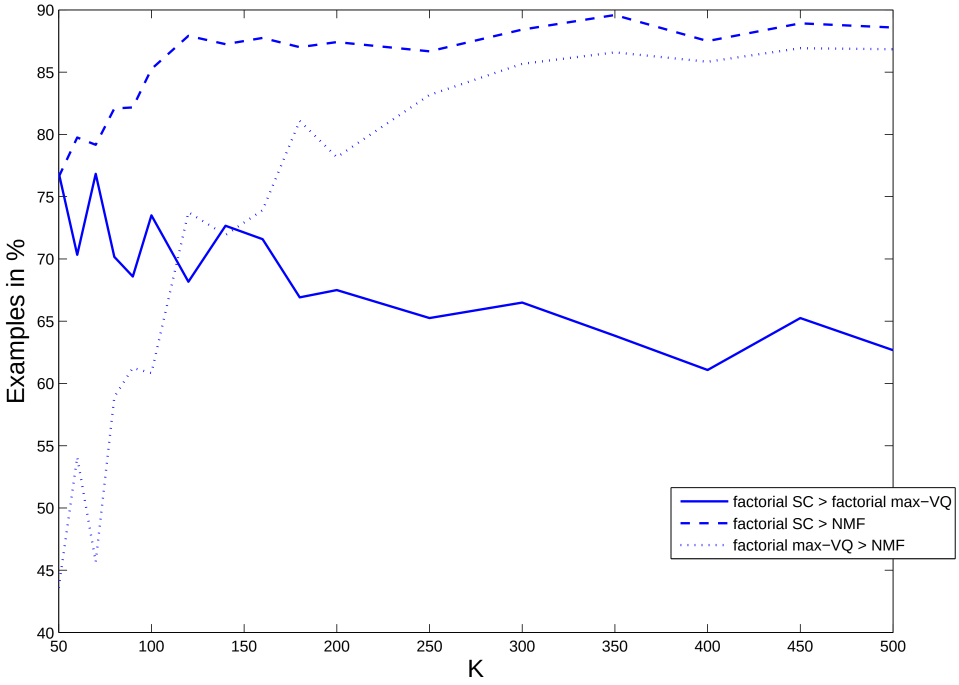


Figure 4.14: Percentage of examples where factorial SC performed better than factorial max-VQ (solid), factorial SC performed better than NMF (dashed) and factorial max-VQ performed better than NMF (dotted).

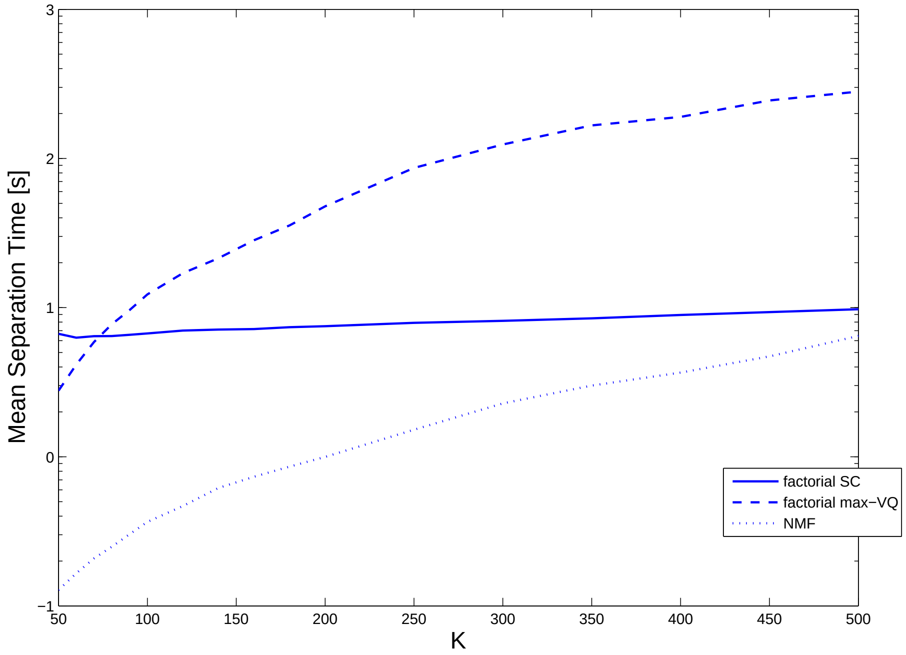


Figure 4.15: Computational effort for factorial SC (solid), factorial max-VQ (dashed) and NMF (dotted). The ordinate is in logarithmic scale (base 10).

Further, we measured the time which was needed for the method specific inference. Figure 4.15 compares the computational effort as a function of the dictionary size. The effort for the factorial SC inference is nearly constant, since it mainly depends on the BS-NMP parameters M and T, which remain fixed. For $K = 500$, the time needed for factorial SC is approximately 10 seconds, which is more than 15 times faster than max-VQ. The time needed by max-VQ grows nearly quadratically, since in principle a full search has to be performed.

4.4 Speech Recognition

In the experiments carried out so far we only used artificial speech mixtures, which were generated by simply adding the source signals. For a more realistic setup we applied the factorial SC system to data from the “SAIL real life SCSS corpus” [10], which provides clean speech of several persons from TV broadcasts. Additionally, it contains recordings of situations where two of these persons interfere. We applied our system to these real-world mixture data, after training dictionaries for the individual speakers. We used approximately 2 minutes of speech for training, the dictionary size was set to $K = 300$ and the number of allowed atoms was set to $L = 3$. Note that this task is particularly challenging, since the training data usually contains read text, while the mixture recordings contain spontaneous dialog and sometimes even laughter. Nevertheless we found in informal listening tests that our system successfully filtered out the target person, while suppressing the interfering speech. Notable artifacts were a vocoder like noise and occasional distortions of the target voice. To quantify the separation quality, we compared the results of an automatic speech recognizer (ASR) (SAIL LABS Media Mining Indexer version 5.1 (MMI) [34]), where one time the mixtures and the other time the demixed signals were used as input. For this experiments we used 13 utterances of 4 different speakers: Alexander von Thien (AvT), Peter Klöppel (PK), Ulrike von der Gröben (UvG) and an unknown female speaker (fu). The utterances are labeled with the speaker id of the target speaker and an increasing counter. The resulting word error rates (WER) of the ASR for the mixed and the demixed signals are compared in Table (4.2).

Utterance	WER mixed	WER demixed
AvT_1	87.5	87.5
AvT_2	44.4	44.4
PK_1	33.3	44.4
PK_2	90	90
PK_3	93.3	80
PK_4	30.8	30.8
UvG_1	80	80
UvG_2	50	33.3
UvG_3	55	70
UvG_4	93.3	86.7
UvG_5	85.7	92.9
fu_1	50	0
fu_2	60	20

Table 4.2: Results of the ASR for mixed and demixed speech.

We see that the source separation step does not significantly decrease the WER for the first three speakers (AvT, PK, UvG). In 4 cases the WER remains unchanged for the demixed signals, in 4 cases it slightly increases and in 4 cases it slightly decreases. For the fourth speaker (fu) however, we see that the source separation preprocessing clearly enhances the result. The mixture utterances of the first three speakers are chatting dialogs, sometimes interleaved with laughter. It seems that these utterances are generally hard to recognize, and the source separation step merely leads to slight variations in the recognition results. On the other hand, the utterances of the fourth speaker are very clear and understandable, and similar to read text. The source separation step delivered a very good separation of target and interfering speaker in this case, what explains the increased performance of the ASR.

Chapter 5

Conclusion

In this thesis we proposed an alternative to nonnegative K-SVD [12], an algorithm for the design of dictionaries for sparse coders. We showed that nonnegative matrix factorization (NMF) as proposed by Lee and Seung [8] is the natural method for the dictionary update step. Further, we proposed a nonnegative version of orthogonal matching pursuit [13], the so called nonnegative matching pursuit (NMP), which can be used for the sparse coding step. Further, we introduced a generalization of the quick but greedy matching pursuit called beam search NMP (BS-NMP). This algorithm allows to consider more than just one atom in each selection step and postpones the selection decision to a later point in time. Our proposed dictionary design algorithm combines NMF with a nonnegative sparse coder which ensures ℓ^0-constraints on the columns of the coefficient matrix. Therefore, we called this method NMFℓ^0.

We applied K-SVD and NMFℓ^0 in a simple method for single channel source separation (SCSS) in order to gain approximations of the original magnitude source spectrograms. The approximations allowed us to estimate masking signals, which can be used for re-filtering [5]. In all separation experiments we received better results using a continuous mask than when a binary mask was used. We observed that the simple system successfully separated sources, where the separation quality did not depend on whether K-SVD or NMFℓ^0 dictionaries were used. However, compared to the factorial max-VQ system by Roweis [6], the performance was not satisfying. Especially in the case of two speakers of the same gender we observed a dramatic performance drop-off. Since the spectra of two same gender speakers overlap to a larger extent, we concluded that the sparse coder model has to be constrained in order to infer spectrogram approximations of higher quality. This conclusion was confirmed by using BS-NMP in the simple system, which results in better error reduction. Although the error was successfully reduced, the separation quality increased only to a small extent. Also the problem of the same gender case remained.

In an optimal case analysis we found upper bounds for the separation quality which can be achieved by max-VQ, K-SVD and NMFℓ^0. We demonstrated, that the more general methods K-SVD and NMFℓ^0 are at least able to perform better than max-VQ. In the next step, we replaced the optimal coefficients with values found by a least squares approximation of the data, using the atoms selected by the optimal case analysis. The separation performance after this experiment was superior for NMFℓ^0. We concluded that an inference method for NMFℓ^0 has to provide merely the indices of suitable atoms in order to achieve good separation quality.

Finally, we proposed a probabilistic factorial sparse coder (SC) model. We defined the posterior probability of atom selections for each source, given the mixture data. To find the optimal atom selection is intractable, therefore we proposed to search over the space which is considered in BS-NMP. In experiments, we observed a superior performance of the factorial SC system in comparison to the baseline system, namely factorial max-VQ. We observed that the standard deviation of the SIR is correlated with the achieved mean value. However, we can state that factorial SC achieves in more than 60 % of the cases a better result than factorial max-VQ.

Measurements of the method specific inference times show that our system is up to 15 times faster than factorial max-VQ.

5.1 Future Work

The discussed models need to be provided with clean, speaker specific data. This is unsatisfying, since in real applications we can not provide training data for every unknown source. Therefore we suggest to modify the discussed systems in order to be more general and less source specific. For instance, we could train a dictionary on a variety of sounds, and further organize this dictionary in order to be able to produce source specific dictionaries on the fly.

Although NMF seems to be the optimal choice for the dictionary update step in nonnegative dictionary design, the problem of nonnegative sparse coding is not solved to a satisfying extend.

Finally, the factorial sparse coder model could be refined by additional constraints on the coefficients *values*. We showed that we can achieve good results without taking these into account, and therefore we could simplify our task. However, to incorporate prior knowledge about the coefficient values could further increase the separation quality.

Bibliography

[1] Albert S. Bregman, *Auditory Scene Analysis: The perceptual Organization of Sound*, MIT Press, Cambridge, MA, 1999 2nd pp. edition, 1990.

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

[3] D. L. Wang and G. J. Brown, "Separation of speech from interfering sounds based on oscillatory correlation," *IEEE Trans. on Neural Networks*, vol. 10, no. 3, pp. 684–697, 1999.

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

[5] S.T. Roweis, "One microphone source separation," in *Neural Information Processing Systems*, 2001, pp. 793–799.

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

[7] M. Aharon, M. Elad, and A.M. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," in *IEEE Transactions on Signal Processing*, 2006, vol. 54, pp. 4311–4322.

[8] 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.

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

[10] Riedler J. and Stark M., "A real life corpus for single channel source separation," Tech. Rep., Graz University of Technology, SPSC Lab.; SAIL LABS Technology AG, 2008.

[11] E.C Cherry, "Some experiments on the recognition of speech, with one and with two ears," *Journal of the Acoustic Society of America*, vol. 25, pp. 975–979, 1953.

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

[13] J.A. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

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

[15] A. J. Bell and T. J. Sejnowski, "An information-maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[16] R. Linsker, "An application of the principle of maximum information preservation to linear systems," *Advances in neural information processing systems*, vol. 1, pp. 186 – 194, 1989.

[17] R. Linsker, "Local synaptic learning rules suffice to maximize mutual information in a linear network," *Neural Computation*, vol. 4, pp. 691–702, 1992.

[18] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters.," *Vision Research*, vol. 37, pp. 3327–3338, 1997.

[19] B.A. Oldhausen and D.J. Field, "Sparse coding with an overcomplete basis set: a strategy employed by v1?," *Vision Res*, vol. 37, pp. 3311–3325, 1997.

[20] D.J. Field, "What is the goal of sensory coding?," *Neural Computation*, vol. 6, pp. 559–601, 1994.

[21] D.J. Willshaw, O.P Buneman, and H.C. Longuet-Higgins, "Non-holographic associative memory," *Nature*, vol. 222, pp. 960 – 962, 1969.

[22] G. Palm, "On associative memory," *Biological Cybernetics*, vol. 36, pp. 19–31, 1980.

[23] P. Fldik, "Forming sparse representations by local anti-hebbian learning," *Biological Cybernetics*, vol. 64, pp. 165–170, 1990.

[24] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[25] Patrik O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, 2004.

[26] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *International Computer Music Conference*, International Computer Music Conference, ICMC.

[27] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1998.

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

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

[30] B.D. Rao and K. Kreutz-Delgado, “An affine scaling methodology for best basis selection,” in *IEEE Transactions on Signal Processing*, 1999.

[31] R. Rubinstein, M. Zibulevsky, and M. Elad, “Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit,” Tech. Rep., Technical Report - CS Technion, 2008.

[32] C.L Lawson and R.J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, 1974.

[33] A. Y. Yang, S. Maji, K. Hong, P. Yan, and S. S. Sastry, “Distributed compression and fusion of nonnegative sparse signals for multiple-view object recognition,” in *International Conference on Information Fusion*, 2009.

[34] Mark Pfeiffer, Marco Avila, Gerhard Backfried, Norbert Pfannerer, and Juergen Riedler, “Next generation data fusion open source intelligence (osint) system based on mpeg7,” in *IEEE International Conference on Technologies for Homeland Security*, 2008.

Appendix A

Derivations

A.1 Convolution of Two Laplacian Distributions

We want to show that the convolution of the two Laplacian distributions

$$
\begin{aligned}
p_1(x|\mu_1,\lambda_1) &= \tfrac{1}{2\lambda_1} e^{\frac{-|x-\mu_1|}{\lambda_1}} && \text{(A.1)}\\
p_2(x|\mu_2,\lambda_2) &= \tfrac{1}{2\lambda_2} e^{\frac{-|x-\mu_2|}{\lambda_2}} && \text{(A.2)}
\end{aligned}
$$

is given as

$$
\begin{aligned}
p(x|\mu_1,\lambda_1,\mu_2,\lambda_2) &= p_1(x|\mu_1,\lambda_1) * p_2(x|\mu_2,\lambda_2) && \text{(A.3)}\\
&= \frac{1}{2}\left[\left(\frac{\lambda_1}{\lambda_1^2-\lambda_2^2}\right) e^{\frac{-|x-(\mu_1+\mu_2)|}{\lambda_1}} + \left(\frac{\lambda_2}{\lambda_2^2-\lambda_1^2}\right) e^{\frac{-|x-(\mu_1+\mu_2)|}{\lambda_2}}\right] && \text{(A.4)}
\end{aligned}
$$

Firstly, consider two zero-mean Laplacian distributions $p_{Z1}(x|\lambda_1)$ and $p_{Z1}(x|\lambda_2)$:

$$
\begin{aligned}
p_{Z1}(x|\lambda_1) &= \tfrac{1}{2\lambda_1} e^{\frac{-|x|}{\lambda_1}} && \text{(A.5)}\\
p_{Z1}(x|\lambda_2) &= \tfrac{1}{2\lambda_2} e^{\frac{-|x|}{\lambda_2}} && \text{(A.6)}
\end{aligned}
$$

The convolution $p_Z(x|\lambda_1,\lambda_2)$ of $p_{Z1}(x|\lambda_1)$ and $p_{Z2}(x|\lambda_2)$ is defined as:

$$
\begin{aligned}
p_Z(x|\lambda_1,\lambda_2) &= (p_{Z1} * p_{Z2})(x|\lambda_1,\lambda_2)\\
&= \int_{-\infty}^{\infty} p_{Z1}(y|\lambda_1)\, p_{Z2}(x-y|\lambda_2)\, dy\\
&= \int_{-\infty}^{\infty} \frac{1}{2\lambda_1} e^{\frac{-|y|}{\lambda_1}} \; \frac{1}{2\lambda_2} e^{\frac{-|x-y|}{\lambda_2}} \; dy && \text{(A.7)}
\end{aligned}
$$

To solve the integral in Eq. (A.7), we have to consider two cases:

1. $x < 0$
2. $x \geq 0$

Case 1: $x < 0$

$$
\begin{aligned}
p_Z(x|\lambda_1,\lambda_2) &= \int_{-\infty}^{\infty} \frac{1}{2\lambda_1}\, e^{\frac{-|y|}{\lambda_1}}\; \frac{1}{2\lambda_2}\, e^{\frac{-|x-y|}{\lambda_2}}\, dy \\
&= \frac{1}{4\lambda_1\,\lambda_2}\left[\int_{-\infty}^{x} e^{\frac{y}{\lambda_1}}\, e^{\frac{y-x}{\lambda_2}}\, dy + \int_{x}^{0} e^{\frac{y}{\lambda_1}}\, e^{\frac{x-y}{\lambda_2}}\, dy + \int_{0}^{\infty} e^{\frac{-y}{\lambda_1}}\, e^{\frac{x-y}{\lambda_2}}\, dy\right] \\
&= \frac{1}{4\lambda_1\,\lambda_2}\left[\int_{-\infty}^{x} e^{-\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\, e^{\frac{\lambda_1+\lambda_2}{\lambda_1\,\lambda_2}\,y}\, dy\right. \\
&\quad + \int_{x}^{0} e^{\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\, e^{\frac{\lambda_2-\lambda_1}{\lambda_1\,\lambda_2}\,y}\, dy \\
&\quad \left. + \int_{0}^{\infty} e^{\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\, e^{\frac{-\lambda_1-\lambda_2}{\lambda_1\,\lambda_2}\,y}\, dy\right] \\
&= \frac{1}{4\lambda_1\,\lambda_2}\left[e^{-\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\,\frac{\lambda_1\,\lambda_2}{\lambda_1+\lambda_2}\, e^{\frac{\lambda_1+\lambda_2}{\lambda_1\,\lambda_2}\,y}\right|_{-\infty}^{x} \\
&\quad + e^{\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\,\frac{\lambda_1\,\lambda_2}{\lambda_2-\lambda_1}\, e^{\frac{\lambda_2-\lambda_1}{\lambda_1\,\lambda_2}\,y}\Bigg|_{x}^{0} \\
&\quad + e^{\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\,\frac{\lambda_1\,\lambda_2}{-\lambda_1-\lambda_2}\, e^{\frac{-\lambda_1-\lambda_2}{\lambda_1\,\lambda_2}\,y}\left|_{0}^{\infty}\right] \\
&= \frac{1}{4\lambda_1\,\lambda_2}\left[\frac{\lambda_1\,\lambda_2}{\lambda_1+\lambda_2}\, e^{\frac{\lambda_2}{\lambda_1\,\lambda_2}\,x}\right. \\
&\quad + \frac{\lambda_1\,\lambda_2}{\lambda_2-\lambda_1}\left(e^{\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x} - e^{\frac{\lambda_2}{\lambda_1\,\lambda_2}\,x}\right) \\
&\quad \left. + \frac{\lambda_1\,\lambda_2}{\lambda_1+\lambda_2}\, e^{\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\right] \\
&= \frac{1}{4}\left[\left(\frac{1}{\lambda_1+\lambda_2} - \frac{1}{\lambda_2-\lambda_1}\right) e^{\frac{x}{\lambda_1}} + \left(\frac{1}{\lambda_1+\lambda_2} + \frac{1}{\lambda_2-\lambda_1}\right) e^{\frac{x}{\lambda_2}}\right] \\
&= \frac{1}{4}\left[\left(\frac{2\lambda_1}{\lambda_1^2-\lambda_2^2}\right) e^{\frac{x}{\lambda_1}} + \left(\frac{2\lambda_2}{\lambda_2^2-\lambda_1^2}\right) e^{\frac{x}{\lambda_2}}\right] \\
&= \frac{1}{2}\left[\left(\frac{\lambda_1}{\lambda_1^2-\lambda_2^2}\right) e^{\frac{x}{\lambda_1}} + \left(\frac{\lambda_2}{\lambda_2^2-\lambda_1^2}\right) e^{\frac{x}{\lambda_2}}\right] \qquad \text{(A.8)}
\end{aligned}
$$

Case 2: $x \geq 0$

$$
\begin{aligned}
p_Z(x|\lambda_1,\lambda_2) &= \int_{-\infty}^{\infty} \frac{1}{2\lambda_1}\, e^{\frac{-|y|}{\lambda_1}}\; \frac{1}{2\lambda_2}\, e^{\frac{-|x-y|}{\lambda_2}}\, dy \\
&= \frac{1}{4\lambda_1\,\lambda_2}\left[\int_{-\infty}^{0} e^{\frac{y}{\lambda_1}}\, e^{\frac{y-x}{\lambda_2}}\, dy + \int_{0}^{x} e^{\frac{-y}{\lambda_1}}\, e^{\frac{y-x}{\lambda_2}}\, dy + \int_{x}^{\infty} e^{\frac{-y}{\lambda_1}}\, e^{\frac{x-y}{\lambda_2}}\, dy\right] \\
&= \frac{1}{4\lambda_1\,\lambda_2}\left[\int_{-\infty}^{0} e^{-\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\, e^{\frac{\lambda_1+\lambda_2}{\lambda_1\,\lambda_2}\,y}\, dy\right. \\
&\quad + \int_{0}^{x} e^{-\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\, e^{\frac{\lambda_1-\lambda_2}{\lambda_1\,\lambda_2}\,y}\, dy \\
&\quad \left. + \int_{x}^{\infty} e^{\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\, e^{\frac{-\lambda_1-\lambda_2}{\lambda_1\,\lambda_2}\,y}\, dy\right]
\end{aligned}
$$

$$
\begin{aligned}
= &\ \frac{1}{4\lambda_1\,\lambda_2}\left[e^{-\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\ \frac{\lambda_1\,\lambda_2}{\lambda_1+\lambda_2}\ e^{\frac{\lambda_1+\lambda_2}{\lambda_1\,\lambda_2}\,y}\right|_{-\infty}^{0} \\
&\quad + e^{-\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\ \frac{\lambda_1\,\lambda_2}{\lambda_1-\lambda_2}\ e^{\frac{\lambda_1-\lambda_2}{\lambda_1\,\lambda_2}\,y}\Big|_{0}^{x} \\
&\quad \left. + \ e^{\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\ \frac{\lambda_1\,\lambda_2}{-\lambda_1-\lambda_2}\ e^{\frac{-\lambda_1-\lambda_2}{\lambda_1\,\lambda_2}\,y}\right|_{x}^{\infty}\Bigg] \\
= &\ \frac{1}{4\lambda_1\,\lambda_2}\left[\frac{\lambda_1\,\lambda_2}{\lambda_1+\lambda_2}\ e^{-\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\right. \\
&\quad +\frac{\lambda_1\,\lambda_2}{\lambda_1-\lambda_2}\left(e^{-\frac{\lambda_2}{\lambda_1\,\lambda_2}\,x}-e^{-\frac{\lambda_1}{\lambda_1\,\lambda_2}\,x}\right) \\
&\quad \left. + \ \frac{\lambda_1\,\lambda_2}{\lambda_1+\lambda_2}\ e^{-\frac{\lambda_2}{\lambda_1\,\lambda_2}\,x}\right] \\
= &\ \frac{1}{4}\left[\left(\frac{1}{\lambda_1+\lambda_2}+\frac{1}{\lambda_1-\lambda_2}\right)e^{\frac{-x}{\lambda_1}}+\left(\frac{1}{\lambda_1+\lambda_2}-\frac{1}{\lambda_1-\lambda_2}\right)e^{\frac{-x}{\lambda_2}}\right] \\
= &\ \frac{1}{4}\left[\left(\frac{2\lambda_1}{\lambda_1^2-\lambda_2^2}\right)e^{\frac{-x}{\lambda_1}}+\left(\frac{2\lambda_2}{\lambda_2^2-\lambda_1^2}\right)e^{\frac{-x}{\lambda_2}}\right] \\
= &\ \frac{1}{2}\left[\left(\frac{\lambda_1}{\lambda_1^2-\lambda_2^2}\right)e^{\frac{-x}{\lambda_1}}+\left(\frac{\lambda_2}{\lambda_2^2-\lambda_1^2}\right)e^{\frac{-x}{\lambda_2}}\right]
\end{aligned} \tag{A.9}
$$

We see that we can combine Eq. (A.8) and Eq. (A.9) to

$$
p_Z(x|\lambda_1,\lambda_2)=\frac{1}{2}\left[\left(\frac{\lambda_1}{\lambda_1^2-\lambda_2^2}\right)e^{\frac{-|x|}{\lambda_1}}+\left(\frac{\lambda_2}{\lambda_2^2-\lambda_1^2}\right)e^{\frac{-|x|}{\lambda_2}}\right] \tag{A.10}
$$

We can reformulate $p_1(x|\mu_1,\lambda_1)$ and $p_2(x|\mu_2,\lambda_2)$ as the convolution of the zero-mean pdfs $p_{Z1}(x|\lambda_1)$ and $p_{Z2}(x|\lambda_2)$ with Dirac impulses at the positions μ_1 and μ_2, respectively:

$$
p_1(x|\mu_1,\lambda_1) \quad = p_{Z1}(x|\lambda_1) * \delta(x-\mu_1) \tag{A.11}
$$

$$
p_2(x|\mu_2,\lambda_2) \quad = p_{Z2}(x|\lambda_2) * \delta(x-\mu_2) \tag{A.12}
$$

Using Eq. (A.10), (A.11) and (A.12), we can reformulate $p(x|\mu_1,\lambda_1,\mu_2,\lambda_2)$ as

$$
\begin{aligned}
p(x|\mu_1,\lambda_1,\mu_2,\lambda_2) &= p_1(x|\mu_1,\lambda_1) * p_2(x|\mu_2,\lambda_2) \\
&= p_{Z1}(x|\lambda_1) * \delta(x-\mu_1) * p_{Z2}(x|\lambda_2) * \delta(x-\mu_2) \\
&= (p_{Z1}(x|\lambda_1) * p_{Z2}(x|\lambda_2)) * (\delta(x-\mu_1) * \delta(x-\mu_2)) \\
&= p_Z(x|\lambda_1,\lambda_2) * \delta(x-\mu_1-\mu_2) \\
&= \frac{1}{2}\left[\left(\frac{\lambda_1}{\lambda_1^2-\lambda_2^2}\right)e^{\frac{-|x-(\mu_1+\mu_2)|}{\lambda_1}}+\left(\frac{\lambda_2}{\lambda_2^2-\lambda_1^2}\right)e^{\frac{-|x-(\mu_1+\mu_2)|}{\lambda_2}}\right]
\end{aligned}
$$

□