Single-channel source separation using non-negative matrix factorization

Mikkel N. Schmidt

Kongens Lyngby 2008
IMM-PHD-2008-x

Technical University of Denmark
Informatics and Mathematical Modeling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk

IMM-PHD: ISSN 0909-30192

Summary

Single-channel source separation problems occur when a number of sources emit signals that are mixed and recorded by a single sensor, and we are interested in estimating the original source signals based on the recorded mixture. This problem, which occurs in many sciences, is inherently underdetermined and its solution relies on making appropriate assumptions concerning the sources.

This dissertation is concerned with model-based probabilitic single-channel source separation based on non-negative matrix factorization, and consists of two parts: i) three introductory chapters and ii) five published papers. The first part introduces the single-channel source separation problem as well as non-negative matrix factorization and provides a comprehensive review of existing approaches, applications, and practical algorithms. This serves to provide context for the second part, the published papers, in which a number of methods for single-channel source separation based on non-negative matrix factorization are presented. In the papers, the methods are applied to separating audio signals such as speech and musical instruments and separating different types of tissue in chemical shift imaging.

Resumé

Kildeseparationsproblemer i én kanal opstår når et antal kilder udsender signaler som blandes og optages med én enkelt sensor, og vi er interesseret i at estimere de originale kildesignaler baseret på det optagne mikstursignal. Dette problem, som opstår indenfor mange grene af videnskaberne, har en iboende underbestemthed og dets løsning beror på at indføre passende antagelser om signalkilderne.

Denne afhandling omhandler modelbaseret probabilistisk kildeseparation i én kanal, baseret på ikke-negativ matrix-faktorisering, og består af to dele: i) tre introducerende kapitler og ii) fem publicerede artikler. Den første del introducerer enkeltkanals-kildeseparationsproblemet såvel som ikke-negativ matrix-faktorisering og giver en omfattende redegørelse for eksisterende tilgange, anvendelser og praktiske algoritmer. Dette har til formål at give kontekst til den anden del, de publicerede artikler, hvori et antal metoder til enkeltkanals-kildeseparation baseret på ikke-negativ matrix-faktorisering præsenteres. I artiklerne anvendes metoderne blandt andet til separation af lydsignaler såsom tale og musikinstrumenter samt til separation af forskellige vævstyper i billeddannende spektroskopi.

Preface

This thesis was prepared at Informatics and Mathematical Modelling (IMM), the Technical University of Denmark (DTU) in partial fulfillment of the requirements for acquiring the Ph.D. degree in engineering.

The thesis consists of a summary report and a collection of five research papers written during the period 2005–2008, and elsewhere published.

Lyngby, 2008

Mikkel N. Schmidt

Papers included in the thesis

[A] Mikkel N. Schmidt and Morten Mørup, "Non-negative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA), Lecture Notes in Computer Science (LNCS)*, Springer, vol. 3889, pp. 700–707, Apr. 2006.

[B] Mikkel N. Schmidt and Rasmus K. Olsson, "Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization," in *Spoken Language Processing, ICSLP International Conference on (INTERSPEECH)*, Sep. 2006.

[C] Mikkel N. Schmidt, Jan Larsen, and Fu-Tien Hsiao, "Wind Noise Reduction using Non-negative Sparse Coding," in *Machine Learning for Signal Processing, IEEE International Workshop on (MLSP)*, pp. 431–436, Aug. 2007.

[D] Mikkel N. Schmidt and Rasmus K. Olsson, "Linear Regression on Sparse Features for Single-Channel Speech Separation," in *Applications of Signal Processing to Audio and Acoustics, IEEE Workshop on (WASPAA)*, Oct. 2007.

[E] Mikkel N. Schmidt and Hans Laurberg, "Non-negative Matrix Factorization with Gaussian Process Priors," in *Computational Intelligence and Neuroscience*, May 2008.

Acknowledgements

I wish to express my deepest gratitiude to my supervisor, Professor Jan Larsen, for his continuous encouragement and support. I would like to thank past and present members of the Intelligent Signal Processing group, especially Professor Lars Kai Hansen, for creating an inspiring and cheerful environment.

I warmly thank my co-authors, Morten Mørup, Rasmus K. Olsson, Jan Larsen, Fu-Tien Hsiao, and Hans Laurberg, for the pleasure of their collaboration and useful discussions.

During my studies I spent six months at Columbia University. I would like to thank Professor Dan Ellis for his hospitality and kind advice, and everybody at LabROSA for making my stay unforgettable. I gratefully acknowledge the support from Oticon Fonden, Otto Mønsteds Fond, and Marie & M.B.Richters Fond for supporting my stay at Columbia University.

Abbreviations

AR	Auto-regressive
ASR	Automated speech recognition
BFGS	Broyden-Fletcher-Goldfarb-Shanno
BSS	Blind source separation
CANDECOMB	Canonical decomposition
CASA	Computational auditory scene analysis
DCT	Discrete cosine transform
EEG	Electroencephalogram
GMM	Gaussian mixture model
GPP	Gaussian process prior
HMM	Hidden Markov model
ICA	Independent component analysis
KKT	Karush-Kuhn-Tucker
KL	Kullback-Leibler
L-BFGS-B	Limited-memory BFGS with bound constraints
LPC	Linear predictive coefficient
LP	Low-pass
LSA	Latent semantic analysis
LS	Least squares
MAP	Maximum a posteriori
MCMC	Markov chain Monte Carlo

MFCC	Mel frequency cepstral coefficient
ML	Maximum likelihood
MMAP	Marginal maximum a posteriori
MMSE	Minimum mean square error
NMF2D	Non-negative matrix factor 2-D deconvolution
NMFD	Non-negative matrix factor deconvolution
NMF	Non-negative matrix factorization
NNLS	Non-negative least squares
NNSC	Non-negative sparse coding
NTF	Non-negative tensor factorization
PARAFAC	Parallel factor analysis
PCA	Principal component analysis
PET	Positron emission tomography
PLSA	Probabilistic latent semantic analysis
PM	Posterior mean
PMF	Positive matrix factorization
RVM	Relevance vector machine
SNR	Signal-to-noise ratio
SOCP	Second order cone programming
STFT	Short time Fourier transform
SVD	Singular value decomposition
SVM	Support vector machine
VQ	Vector quantization
WLS	Weighted least squares
WP	Weighted Poisson

Contents

Chapter 1

Introduction

Source separation problems arise when a number of sources emit signals that mix and propagate to one or more sensors. The objective is to identify the underlying source signals based on measurements of the mixed sources. This thesis deals with the underdetermined problem of source separation when the mixed signals are recorded using only a single sensor.

Source separation methods can be divided into blind and non-blind methods. Blind source separation (BSS) denotes the separation of completely unknown sources without using additional information. BSS methods typically rely on the assumption that the sources are non-redundant, and the methods are based on, for example, decorrelation, statistical independence, or the minimum description length principle. Non-blind source separation denotes the separation of sources for which further information is available, for example in terms of a prior distribution. The single-channel source separation problem is underdetermined and cannot in general be solved using completely blind methods.

Sometimes separating a single-channel mixture of sources is easy, because some simple natural characteristic can be used to distinguish the sources. This is the case, for example, when the sources lie in known disjoint frequency bands. When no such simple natural characteristic separates the sources, the problem can be extremely difficult.

The single-channel source separation problem is ubiquitous in many different application areas including:

Audio processing, for example to separate instruments in music recordings [A, 208, 224, 225], to separate the voices of multiple simultaneous speakers [B, D, 122, 223], or to reduce background noise [C];

Bioinformatics, for example to identify and discriminate between different types of tissue in chemical shift imaging [E, 151, 199, 200];

Chemometrics, for example to determine the spectra and concentration profiles of chemical components in an unresolved mixture [73];

Environmetrics, for example to identify the sources of pollutant particles in spectroscopic measurements of air quality [129]; and

Image processing, for example to extract meaningful image features or separate mixed images [140].

1.1 Thesis outline and contributions

The focus of this thesis is model-based probabilistic separation of single channel recordings of mixed sources using non-negative matrix factorization. The thesis consists of two introductory chapters and five published papers that constitute the main contribution of the thesis. The introductory chapters review existing methods for probabilistic single channel source separation and non-negative matrix factorization respectively, and the aim is to give an overview of the field and place the published papers into context.

Chapter 2, Single-channel source separation, introduces the single channel source separation problem, and discusses several approaches to solving the problem. A general framework for model-based separation is presented, and aspects that distinguish different methods are discussed. Focused on model-based probabilistic methods, a comprehensive review of existing approaches to single channel source separation is presented.

Chapter 3, Non-negative matrix factorization, gives an introduction to non-negative matrix factorization (NMF) and presents a probabilistic framework for NMF. A comprehensive review of applications, generalizations and extensions for NMF is provided, and number of computation strategies as well as practical algorithms for NMF are discussed.

Paper A, Non-negative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation, presents a novel method for blind separation of music instruments in a single channel audio recording. The paper introduces a 2-D convolutive extension of NMF, where each instrument is modeled by one basis that is convolutive in time and in frequency to model temporal structure as well as pitch changes. The method is based on a non-negative factorization of a log-frequency spectrogram and exploits that a pitch change corresponds to a displacement on the logarithmic frequency axis. Where previous methods needed one component to model each note for each instrument, the proposed model represents each instrument compactly by a single time-frequency profile convolved in both time and frequency by a time-pitch weight matrix. This model effectively solves the blind single channel source separation problem for certain classes of musical signals.

Paper B, Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization, deals with the separation of multiple speech sources from a single microphone recording. The approach is based on a sparse non-negative matrix factorization, that is used to learn speaker models from a speech corpus. These models are then used to separate the audio stream into its components. We show that considerable computational savings can be achieved by segmenting the training data into phoneme-level subproblems using a speech recognizer.

Paper C, Wind Noise Reduction using Non-negative Sparse Coding, introduces a speaker independent method for reducing wind noise in single-channel recordings of noisy speech. The method is based on sparse non-negative matrix factorization and relies on a noise model that is estimated from isolated noise recordings. The paper compares the proposed method with the classical spectral subtraction method and a state-of-the-art noise reduction method, and shows that the proposed method achieves a considerably improved signal-to-noise ratio.

Paper D, Linear Regression on Sparse Features for Single-Channel Speech Separation, addresses the problem of separating multiple speakers from a single microphone recording by the formulation of a linear regression model, that estimates each speaker based on features derived from the mixture. In the paper, two feature representations are compared: short-time Fourier transform features, and sparse non-negative encoding of the speech mixture computed using sparse NMF. Results show that combining sparse non-negative features with a regression model leads to a significantly improved performance in terms of signal-to-noise ratio.

Paper E, Non-negative Matrix Factorization with Gaussian Process Priors, presents a general method for including prior knowledge in a

non-negative matrix factorization, based on Gaussian process priors. The method is derived in a probabilistic setting, based on specifying prior probability distributions of the factors in the NMF model. It is assumed that that the factors are linked by a strictly increasing function to an underlying Gaussian process, specified by its covariance function, which makes it possible to find NMF decompositions that agree with prior knowledge, such as sparseness, smoothness, and symmetries. Results on a dataset from chemical shift brain imaging show that better spatial separation between spectra corresponding to muscle and brain tissue can be achieved.

Chapter 2

Single-channel source separation

The single-channel source separation problem can be defined as the estimation K original source signals, $s_1(n), \ldots, s_K(n)$, given only an observed mixture, $x(n)$. In a general formulation we may write

$$x(n) = g\big(s_1(n), \ldots, s_K(n)\big), \tag{2.1}$$

where g is some possibly non-linear and stochastic mixing process. Often, the mixing process is taken as the sum of the sources plus additive independent noise, such that

$$x(n) = \sum_{k=1}^{K} s_k(n) + e(n), \tag{2.2}$$

where $e(n)$ is a noise term. The variable n usually denotes time, and will be referred to as such in the following, but depending on the application n could also represent space, frequency, wavenumber, etc. When the signal can be represented by a discrete sample, we may write (2.1) in vector notation as $\boldsymbol{x} = g(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K)$, where $\boldsymbol{s}_k = [s_k(1), \ldots, s_k(N)]^\top$ and g is taken element-wise.

Single-channel source separation is an underdetermined problem and its solution requires additional information about the sources. For example, it is evident that in the case of linear noise-free mixing with two sources $\boldsymbol{s}_1 = \bar{\boldsymbol{s}}$ and $\boldsymbol{s}_2 = \boldsymbol{x} - \bar{\boldsymbol{s}}$

is a solution for any $\bar{\boldsymbol{s}}$, and it is necessary to use additional information about the sources to constrain the problem. For this reason, the single-channel source separation problem lends itself well to be treated by machine learning methods in a probabilistic framework, where source specific knowledge can be formulated in terms of prior probability distributions, and statistical inference methods can be used to infer the most probable solution to the separation problem.

Several different approaches to single-channel source separation have been proposed in the literature, most of which can be seen as i) filtering, ii) decomposition and grouping, or iii) source modeling approaches.

In the filtering approach, a set of functions (filters) are found that transform the mixture to estimates of the sources. For example, one could use matched linear filters that are optimized to extract a single source and maximize signal-to-noise ratio (SNR). More generally, the transformation functions can be chosen from some parameterized family of functions, and the parameters can be learned from training data.

In the decomposition and grouping approach, the signal mixture is first decomposed into components that are known to scatter the sources. These components are subsequently grouped together to form source estimates. The decomposition into components can for example be achieved through a fixed transformation such as the short-time Fourier transform [11], a physically inspired signal representation as in the computational auditory scene analysis (CASA) literature [63, 100, 101, 182], a general parameterized signal model such as a sinusoidal model [215, 218, 219, 224], or matrix factorization methods such as NMF [226]. The grouping of components into source estimates can be done manually [225], using knowledge-based grouping rules, or by machine learning clustering techniques [90]. In some approaches, parameters of a clustering procedure are learned from training data [9–11].

In the source modeling approach, a statistical model is formulated for each of the sources as well as for the mixing process. Model parameters are often learned for the source models using training data, and the sources are separated by statistical inference in the joint model.

The focus of this thesis is on source modeling approaches. Section 2.1 gives a general introduction to model-based probabilistic source separation, and section 2.2 reviews a number of different approaches presented in the literature.

2.1 Model-based probabilistic source separation

In model-based probabilistic source separation, probabilistic models are defined for the sources as well as for the mixing process. The unknown sources are treated as stochastic variables, and the source separation problem is solved by making inference in the joint model.

A general framework for model-based statistical single-channel source separation is illustrated in Figure 2.1. The input to the source separation system is the mixed signal, $x(n)$, and for supervised methods also training data for some or all of the source signals. The mixture is first transformed into an appropriate representation, in which the signal separation is performed. The source models are either constructed directly based on knowledge of the signal sources, or by learning from training data. In the inference stage, the models and data are combined to yield estimates of the sources, either directly or through a signal reconstruction step.

Differences between various model-based single-channel source separation methods can be seen as different choices of signal representation, mixing and source models, method of inference, and signal reconstruction technique, which is is discussed in the following sections.

2.1.1 Signal representation

Source separation is often not computed directly in the original representation of the recorded signal; rather, the signal is transformed to some other representation that is, e.g., chosen to accomplish the following:

Emphasize desired characteristics. Signal representations can be chosen in order to accentuate important characteristics in the signal that helps discriminate between sources. For example, transformations such as the Fourier transform, discrete cosine transform, and wavelet transforms are useful when the source signals are sparse in the transformed domain, and this can lead to simpler separation algorithms: When sources are disjoint[1] in the transformed domain, perfect separation can be achieved by a binary mask. Another example is perceptually weighted time-frequency representations, that are often used in audio separation, where the perceptually most important characteristics are emphasized.

[1] This is related to the concept of *W-disjoint orthogonality* [107, 194, 195], i.e., sources with disjoint support in the STFT domain.

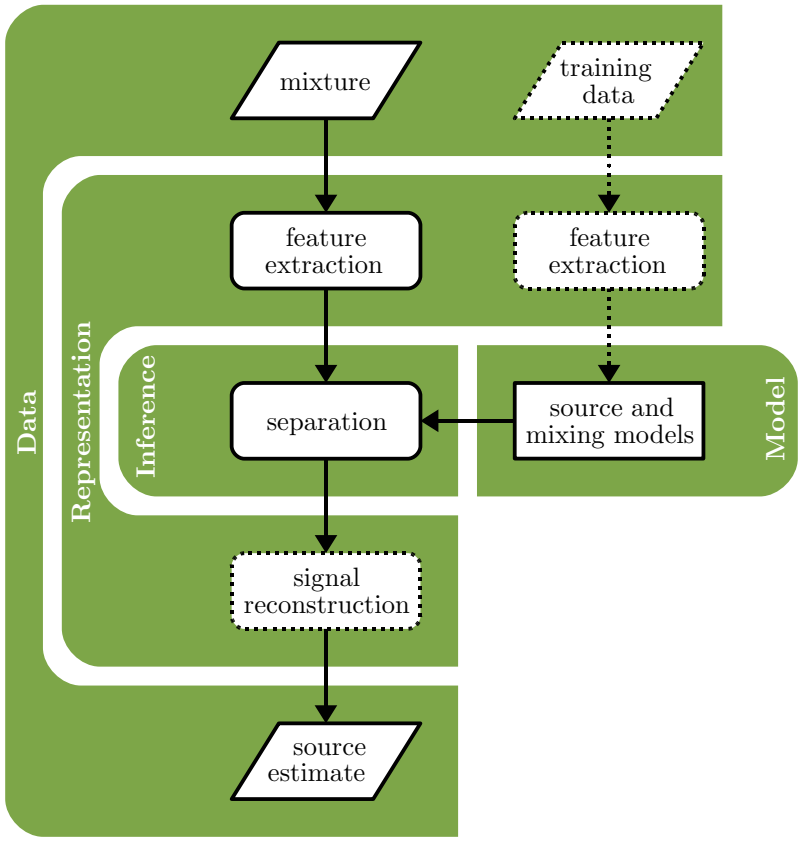

Figure 2.1: A general framework for model-based statistical single-channel source separation. Input to the separation system is the signal mixture and possible training data for the source models. The mixture is transformed into a suitable representation and combined with the source models and mixing model in the inference stage, that either directly or through a signal reconstruction method computes estimates of the separated sources.

Introduce invariances. In addition to accentuating important characteristics, the signal representation can also be chosen to diminish adverse characteristics, that are known to be unimportant for separating the signals. For example, many source separation methods (for example [A,D]) use a power or amplitude spectral representation that disregards the phase, which leads to an invariance to phase shift. Some speech separation methods (for example [223]) are based on Mel-frequency cepstral coefficient (MFCC) features, and when only low-quefrency features are retained an invariance to pitch is introduced. Introducing invariances in the representation can be very helpful for source separation methods based on generative models, since there may be no point in modeling characteristics of the sources that are known to be unimportant for separation.

Allow assumptions of independence or exchangeability. It is often useful to model parts of data as independent or exchangeable. The signal can, for example, be divided into (possibly overlapping) blocks, that are treated as independent, exchangeable, or as loosely coupled sub-problems. The assumption that the signal blocks are independent, exchangeable, or perhaps dependent only on the previous block can lead to more efficient methods of inference.

Reduce dimensionality. With the main purpose of reducing computational cost, it can be of interest to reduce the dimension of the data prior to modeling. This can be done, for example, using principal component analysis (PCA) which is the least-squares best linear technique, or using more advanced non-linear techniques[2].

Allow signal reconstruction. An important distinction is between reversible (lossless) and non-reversible (lossy) signal representations. In the former case, the signals may be separated in the representation domain and the representation inverted to yield separated signals in the original signal domain. In the latter case, however, after the signals are separated in the representation domain, separated signals must be reconstructed in the original signal domain. This can be achieved, e.g., using a filtering approach where the source estimate is used to construct a filter that is applied to the signal mixture. This filter can be a time varying Wiener filter [95], binary [101, 197, 212] or soft [189] masking in a transform domain, etc. It is important that the signal representation is chosen such that adequate signal reconstruction is attainable.

[2]For a review of dimension reduction techniques, see [68].

2.1.2 Mixing and source models

The mixing and source models are used to define what we know, and quantify what we do not know, about the mixing process and the signal sources. The mixing and source models are chosen to i) capture properties of the sources and mixing process to effectively allow the sources to be separated, and ii) have a convenient parametric form to allow efficient inference.

Mixing model

The mixing process, denoted by g in (2.1), is specified in terms of a likelihood function, $p(\boldsymbol{x}|\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K)$, that expresses the probability of observing the mixture when the sources are given.

A simple and often used approach is based on linear mixing with additive noise, $\boldsymbol{x} = \sum_k \boldsymbol{s}_k + \boldsymbol{e}$, which gives rise to the likelihood function

$$p(\boldsymbol{x}|\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K) = p_e\left(\boldsymbol{x} - \textstyle\sum_k \boldsymbol{s}_k\right), \tag{2.3}$$

where $p_e(\cdot)$ is the density of the noise. The noise density can be used to model observation noise; in addition to this, when the linear mixing model is used as an approximation to a more involved true mixing process, the noise density can be used as an approximation to non-linearities and cross-terms etc.

Many different mixing models suited for different problems have been proposed in the literature. The mixing model is often chosen in order to trade off the following two objectives:

Accurately model the mixing process. When detailed knowledge about the mixing process is available, a specialized likelihood function can be constructed. Consider, for example, the separation of amplitude spectra. The amplitude spectrum of a mixture is not generally equal to the sum of amplitude spectra of the sources because there may be a phase difference between the sources. If the phase difference is taken into account, e.g., by modeling it as a uniform random variable [163, 164], this gives rise to a likelihood function that is specialized to the separation of amplitude spectra.

Enable efficient inference. Another important consideration for choosing a suitable mixing model is the complexity of making inference in the model. For example, when the sources are modeled by discrete-state models, such as hidden Markov models or vector quantization, inference in the joint model is expensive because of the exponential number of combined states. When the mixing model, however, is chosen as the element-wise maximum of the sources, efficient inference algorithms can be constructed [173, 197, 198], because this effectively decouples the sources in each observation.

Source model

The available a priori knowledge about the sources is specified in terms of a prior distribution, $p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K)$, that factorizes as $\prod_k p(\boldsymbol{s}_k)$ when the source signals are assumed statistically independent.

The priors can be seen as generative models for the sources; however, it is not necessary for the priors to capture all properties of the source distributions for the separation system to be effective. Priors can often be chosen to capture only key characteristics that are sufficient to separate the sources. Similar to the choice of mixing model, the sources models must be chosen to adequately model the sources while allowing efficient inference in the joint model.

The source priors can be chosen by using or combining three levels of source modeling:

Model building. Source models can be chosen based on prior knowledge about the nature of the sources. For example, if the source signals are generated by a physical system, models of the sources can be constructed based on knowledge of the physics.

Model training. When training data is available for the sources, this can be used to create or estimate parameters of the source models. In model training, a suitable flexible parameterized family of models is chosen and parameters of the model are learned from the training data. There are several challenges with respect to model training, such as: i) *Availability of training data*, i.e., are representative isolated recordings available for each source? ii) *Mismatch with training conditions*, i.e., are there external variabilities, for example in the channel or sensors, that cannot be captured in a training set? iii) *Issues of selectivity*, i.e., it may be that the a priori distributions of the sources are wide and overlapping, whereas sources in any observed mixture lie within a small subrange; in this case, training accurate source models may not lead to effective source separation.

Model adaptation. It is possible to adjust the source models with respect to the observed mixed signal, which can be used to overcome some of the challenges with model training. Using this model adaption approach [157, 158], signal models are no longer a priori models of the sources, but adapted to the observed mixture; thus, model adaptation can be seen as part of the inference in the joint model.

2.1.3 Inference

The mixing model, specified by the likelihood function, and the source models, specified by the prior densities, can be combined using Bayes' theorem to yield the posterior distribution of the sources,

$$p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K|\boldsymbol{x}) \propto p(\boldsymbol{x}|\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K) \prod_k p(\boldsymbol{s}_k). \tag{2.4}$$

Inference in the joint model corresponds to estimating the sources based on this posterior density. The marginal posterior of the kth source, that describes the distribution of a single source of interest given the data, is found by integrating the posterior density over the $K-1$ other "nuisance" sources,

$$p(\boldsymbol{s}_k|\boldsymbol{x}) = \int \cdots \int p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K|\boldsymbol{x}) d\boldsymbol{s}_1 \cdots d\boldsymbol{s}_{k-1} d\boldsymbol{s}_{k+1} \cdots d\boldsymbol{s}_K. \tag{2.5}$$

To compute a point estimate of the source, several different estimators can be constructed, each of which has different properties and leads to different inference algorithms. One approach is to compute the posterior mean (PM) or minimum mean square error (MMSE) estimator,

$$\hat{\boldsymbol{s}}_k^{(\mathrm{PM})} = \int \boldsymbol{s}_k p(\boldsymbol{s}_k|\boldsymbol{x}) d\boldsymbol{s}_k, \tag{2.6}$$

which requires integrating the posterior density. If this integral cannot be computed analytically, it may be computed numerically, e.g., using Markov chain Monte Carlo (MCMC) methods [148]. The PM is often the preferred estimate; however, in some situations it might not be appropriate: For example, if the marginal posterior is multimodal the PM lies in the region between the modes, possibly at a point with low posterior density.

Another approach is to compute the joint maximum a posteriori (MAP) estimate,

$$\{\hat{\boldsymbol{s}}_1, \ldots, \hat{\boldsymbol{s}}_K\}^{(\mathrm{MAP})} = \underset{\{\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K\}}{\arg\max}\ p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K|\boldsymbol{x}), \tag{2.7}$$

that maximizes the posterior density over the sources. This approach avoids the sometimes difficult integral required for the PM estimate. Although the MAP estimate by definition lies a high posterior density point, the MAP estimate might not be a good solution—it depends on whether a substantial part of the probability mass lies close to the point of maximum density.

Between these two "extremes" are approaches such as the marginal maximum a posteriori (MMAP) estimator,

$$\hat{\boldsymbol{s}}_k^{(\mathrm{MMAP})} = \underset{\boldsymbol{s}_k}{\arg\max}\, p(\boldsymbol{s}_k|\boldsymbol{x}), \tag{2.8}$$

where the nuisance sources are integrated out, and the marginal MAP estimate is computed for the source of interest. Again, if the integral cannot be computed analytically, MCMC methods can be used [5, 60, 196].

The choice of which of these or possibly other methods of inference to use in a particular problem depends on the data and the model, that are sometimes chosen to make a certain efficient method of inference feasible. Practical inference algorithms often employ (combinations of) analytical and numerical integration, Monte Carlo methods, and constrained optimization methods.

2.2 Approaches to single-channel source separation

Many different model-based probabilistic single-channel source separation methods have been proposed in the literature. In this section, which is organized according to the different types of models, a range of these methods are reviewed.

2.2.1 Fully factorized univariate models

The perhaps most simple source model is based on the assumption that each source, $s(n)$, is independent and identically distributed (i.i.d.) with univariate distribution $p\big(s(n)\big)$, i.e., the prior is fully factorized,

$$p(\boldsymbol{s}) = \prod_n p\big(s(n)\big). \tag{2.9}$$

Based on this model, Hansen and Petersen [84] discuss the separation of linear single-channel mixtures of white sources, and show that for general unimodal distributions the problem is ill-posed and the sources cannot be determined. In some specific situations, however, multi-modal distributions can be effectively separated: For example, a noise-free linear mixture of two binary sources, $s_1(n) \in \{0, a\}$, $s_2(n) \in \{0, b\}$, $a \neq b$, can be perfectly separated because the observation can take only four values, $x(n) = s_1(n) + s_2(n) \in \{0, a, b, a+b\}$, each of which uniquely identifies the values of the underlying sources. In general, however, single channel source separation techniques requires models of more advanced statistics of the sources such as temporal or spectral correlations.

2.2.2 Auto-regressive models

A simple model that can capture temporal correlations in the sources is the auto-regressive (AR) model, $s(n) = \sum_{m=1}^{M} \alpha(m)s(n-m)+\nu(n)$, where $\{\alpha(m)\}_{m=1}^{M}$ is a set of coefficients, M is the order of the AR process, and $\nu(n)$ is a white noise process. Balan et al. [13, 14] demonstrate that for a single-channel mixture of stationary AR sources, the parameters of the AR processes can generically be uniquely identified and the sources separated. With respect to non-stationary sources, however, the identification problem is more difficult. For separating slowly changing non-stationary AR sources, the authors propose to first identify the constituent AR processes for the initial N samples in the signal, and use an on-line adaptive sliding-window method to update the AR processes for each new sample.

2.2.3 Factorial vector quantization

In factorial vector quantization (VQ) the mixed signal is represented as a sequence of vectors, $\{\boldsymbol{x}^{(1)}, \ldots \boldsymbol{x}^{(M_x)}\}$, and isolated training data is required for each source in the mixture. The first step in the factorial VQ procedure is for each source to learn a codebook that consists of N_k code vectors,

$$\mathcal{S}_k = \{\boldsymbol{c}_k^{(1)}, \ldots, \boldsymbol{c}_k^{(N_k)}\}. \tag{2.10}$$

The codebook is learned using k-means or another clustering technique to optimally represent each source vector by a single code vector. Inference in a factorial VQ amounts to finding the combination of codebook vectors for each source that optimally accounts for the data. The maximum likelihood estimate, for example, can be computed as

$$\{z_1^*, \ldots, z_K^*\} = \arg \max_{z_1,\ldots,z_K} p\left(\boldsymbol{x} \middle| \boldsymbol{c}_1^{(z_1)}, \ldots, \boldsymbol{c}_K^{(z_K)}\right), \tag{2.11}$$

where $p(\cdot)$ is the likelihood function and $z_1, \ldots, z_K$ index the codebooks.

Roweis [197] presents a factorial VQ method for separating audio sources in a log-magnitude spectral representation. A naïve implementation would require a search over all combinations of codebook vectors for each source; however, Roweis presents an efficient branch and bound algorithm for an element-wise maximum observation model. In the same spirit, Pontoppidan and Dyrholm [173] propose a fast hierarchical VQ procedure to search for combinations of codebook vectors.

Ellis and Weiss [64] presents a similar approach, where only one VQ is learned for a single source in the mixture. The mixture is then projected onto the

VQ model, effectively treating the separation as a denoising problem. They further extend the approach using a hidden Markov model to capture temporal constraints in terms of transition probabilities between different subsets of the VQ.

Radfar et al. [179, 180] compare different signal representations for VQ based single-channel speech separation: log magnitude spectral vectors; the modulated lapped transform; and pitch and envelope features. They demonstrate that the spectral representation is superior for speaker dependent separation, whereas the pitch and envelope representation is best for speaker independent separation. In [181] they discuss the selection of window size, which in a spectral representation is a trade-off between the assumption of stationarity, that favors short windows, and spectral resolution, that favors long windows. The authors conclude, that slightly longer windows are useful for the task of speech separation as opposed to the window sizes typically used in, e.g., speech coding.

Srinivasan et al. [210, 211] train codebooks for speech and noise in a linear predictive coefficients (LPC) representation. They present an iterative search method, that finds the most likely combination of codebook vectors by alternating search in the speech and noise codebook. As an alternative to having one, possibly huge, codebook to represent many different types of noise, they propose to learn a set of small codebooks for each type of noise and use a classifier to determine which codebook to use. The authors further propose to first estimate the most likely combination of codebook vectors for the speech and noise, and then improve this estimate by estimating the signal and noise as interpolations between the maximum likelihood vectors and their nearest neighbors. Since interpolation in linear predictive coefficients may be unstable, this is performed in another representation, such as line spectral frequencies.

Blouet et al. [26] compare three codebook based approaches, based on Gaussian scaled mixture models [20], amplitude factor models [19] (non-negative sparse coding), and autoregressive models [210, 211]. The authors conclude, that the autoregressive models effectively captures speech features, whereas the amplitude factor model is better suited for separating music signals.

2.2.4 Gaussian mixture models

A very useful and flexible source model is the Gaussian mixture model (GMM). Here, each source is modeled as a mixture of I (multivariate) Gaussian densities,

$$p(\boldsymbol{s}) = \sum_{i=1}^{I} \pi_i \, \mathcal{N}\left(\boldsymbol{s}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right), \tag{2.12}$$

where $\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-N/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}(\boldsymbol{z}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{z}-\boldsymbol{\mu})\right)$ is the normal density and π_i are mixture coefficients.

In the approach of Beierholm et al. [17], the signal is first partitioned into blocks, and the discrete cosine transform (DCT) is computed for each block. In this representation, the sources are modeled by univariate GMMs that are fully factorized over both time and DCT bands which makes the posterior mean estimator analytically tractable. The parameters of the GMMs are learned from training data for each source, and the method is demonstrated on a mixture of two speech signals. The authors comment, that the method might be improved by explicitly modeling temporal and spectral correlations.

In a related approach, Reddy and Raj [190] use a log-magnitude spectral representation and model each source by a multivariate GMM that captures dependencies across frequency bands. The authors consider an element-wise maximum observation model that makes it possible to derive an analytical expression for the posterior mean estimator. The authors demonstrate the algorithm on a speech separation task, and compare with the factorial VQ approach [197]. Radfar et al. [176, 178] present a similar approach and discuss the use of binary mask signal reconstruction technique as well as a joint source identification and separation procedure. Benaroya et al. [20] (see also [26]) extend the framework to scaled Gaussian mixtures and present a maximum a posteriori (MAP) as well as a posterior mean (PM) estimation technique.

2.2.5 Factorial hidden Markov models

The discrete-state and mixture models discussed in the previous sections represent the mixed signal as a sequence of vectors that are treated as independent problems. The factorial hidden Markov model (HMM) framework [75] extends this by taking into account the dependencies between consecutive vectors. Here, the sources are modeled by independently evolving HMMs, specified by a state transition probability, $p(z_k^{(m)}|z_k^{(m-1)})$ and an emission probability, $p(\boldsymbol{x}|z_1, \ldots, z_K)$, that depends on the state of all HMMs. A graphical model of a two-source factorial HMM is shown in Figure 2.2.

Roweis [198] discusses the use of a factorial HMM with a GMM observation model. In this approach a HMM/GMM is learned for each source on isolated training data, and to separate sources the most likely joint state sequence is inferred. Naïve inference in a factorial HMM is exponential in the number of states of each source-HMM (since the likelihood of all combinations of states must be evaluated) and is only feasible for models with a small number of states; however, Roweis shows that by using an element-wise max observation model,

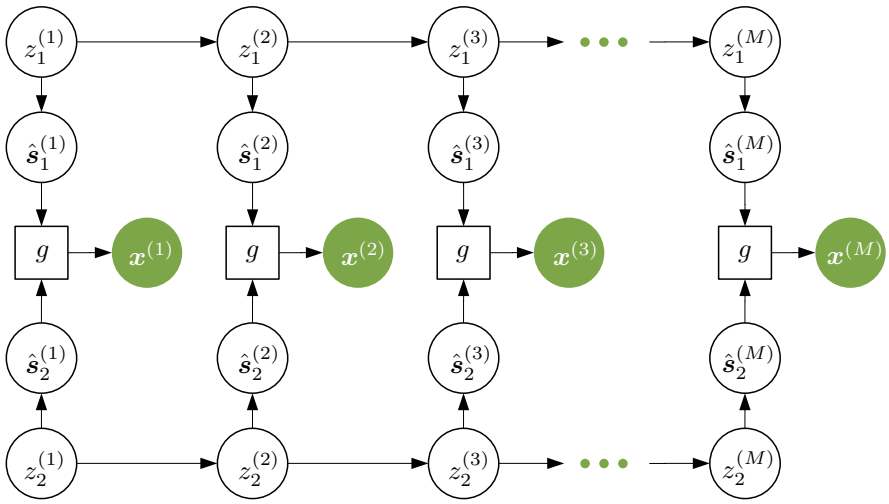

Figure 2.2: Factorial hidden Markov model for two sources. The model consists of two independent hidden Markov models with a combined observation model: $\boldsymbol{x}$ are observed mixtures, g is the observation model, $\hat{\boldsymbol{s}}$ are source estimates, and z are hidden states.

efficient search algorithms exist that makes efficient inference possible even in models with a large number of states. To estimate the separated sources Roweis [198] proposes a re-filtering technique based on a binary mask. Benaroya and Bimbot [18] present a more advanced technique for estimating the sources based on an adaptive Wiener filtering scheme, and Radfar and Dansereau [177] discuss using the MAP estimator.

Kristjansson et al. [122, 229] achieve impressive results on a speech separation problem using an extended HMM/GMM approach in a log power spectral representation. The authors discuss the use of the element-wise max observation model as well as a more advanced model in which exact inference is intractable, and for which an approximation based on Laplace's method is used. For the same problem, Virtanen [223] presents a similar approach that operates in a mel-frequency cepstral coefficient (MFCC) representation and uses a log-normal approximation to make inference in the model tractable.

To accurately model complex sources such as speech in the factorial HMM framework, a very large number of states may be required. Reyes-Gomez et al. [193] present a multiband approach where each source is divided into a number of frequency bands, each of which is modeled by a separate small HMM that is coupled to adjacent bands. Exact inference is intractable in this model because

of the grid-like dependency structure across observations and bands, and the authors present a variational approximation method.

2.2.6 Matrix factorization models

In the matrix factorization approach (also known as latent variable decomposition), sources are modeled by a linear combination of a set of basis vectors,

$$\boldsymbol{s}(n) \approx \sum_{i=1}^{I} \boldsymbol{a}_i b_i(n). \tag{2.13}$$

The basis vectors, $\boldsymbol{a}_i$, that capture the characteristics of the sources, can be learned from isolated training data for each source. The generative model for the sources can be compactly written as a matrix product, $\boldsymbol{S} \approx \boldsymbol{A}\boldsymbol{B}$, where $\boldsymbol{S} = [\boldsymbol{s}(1), \ldots, \boldsymbol{s}(N)]$ is a matrix of N consecutive source vectors, $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_I]$ is a matrix of basis vectors and $\boldsymbol{B} = [\boldsymbol{b}(1), \ldots, \boldsymbol{b}(N)]$, $\boldsymbol{b}(n) = [b_1(n), \ldots, b_I(n)]^\top$ is a matrix of coefficients.

Several different matrix factorization approaches to single-channel source separation have been proposed in the literature. These methods differ by using different matrix factorization techniques for learning the basis, by operating in different signal representation domains, and by relying on different methods of inference.

One of the earliest matrix factorization approaches to single-channel source separation was proposed by Jang et al. [103–106] (see also [102]). In their approach independent component analysis (ICA) is used to learn a set of time-domain basis functions (and coefficient densities). The authors apply the method to different problems in audio source separation and report near-perfect separation when adequate training data is available.

The ICA approach is closely related to the field of sparse coding [134, 154, 155], because the coefficients that are found when the method is applied to, for example, audio signals are sparse, i.e., most of the elements in the coefficient matrix $\boldsymbol{B}$ are zero. When the sources can be represented by a sparse code, it is possible to learn an over-complete [121, 135, 136] basis representation, which is discussed by Pearlmutter and Olsson [169]. In their approach an over-complete basis is learned in a spectral representation using a linear programming technique.

Several authors have proposed using non-negative matrix factorization (NMF) and extensions thereof for learning source bases, based on the assumption that the sources can be meaningfully expressed [8] in a non-negative representation

such as amplitude spectral vectors. NMF can be combined with the idea of sparse coding to form sparse NMF [61, 96–98]. Schmidt and Olsson [B, D] propose to use sparse NMF for source separation by learning an over-complete set of non-negative basis vectors for each source. They show that having a large over-complete basis for representing each source leads to better separation on a speech separation task.

For separating audio sources in a time-frequency representation Smaragdis [208] present a convolutive version of NMF where the bases are time-frequency matrices. This allows the model to capture temporal as well as spectral structure in the sources. In a related approach Schmidt and Mørup [A] propose a 2-D convolutive NMF where convolution in time captures temporal structure and convolution in frequency (on a logarithmic scale) captures pitch change. A similar idea is employed by [114, 115].

Virtanen [220, 224] presents a non-negative sparse coding method that is extended by a continuity objective that allows the model to capture dependencies between source vectors. Schmidt and Laurberg propose to model dependencies between and within source vectors in NMF using a Gaussian process prior [E].

Raj and Smaragdis [183] present a probabilistic latent variable decomposition method that is close in spirit to NMF based source separation. In their approch sources are modeled by a mixture of multinomial distributions. A sparse extension of the approach, proposed by Shashanka et al. [203], allows the computation of an over-complete decomposition. Rennie et al. [192] present a probabilistic framework that includes sparse NMF and mixture-model based source separation as special cases.

Chapter 3

Non-negative matrix factorization

Non-negative matrix factorization (NMF) is a method for approximating a matrix, $\boldsymbol{X}$, as the product of two matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$, under the constraint that all elements in the factorizing matrices be non-negative,

$$\boldsymbol{X} \approx \boldsymbol{AB} \quad \text{s.t. } \boldsymbol{A}, \boldsymbol{B} \geq \boldsymbol{0}, \tag{3.1}$$

where $\boldsymbol{X} \in \mathbb{R}^{I \times J}$, $\boldsymbol{A} \in \mathbb{R}_+^{I \times N}$, and $\boldsymbol{B} \in \mathbb{R}_+^{N \times J}$. In the expression, $\boldsymbol{A}, \boldsymbol{B} \geq \boldsymbol{0}$ means that all elements of $\boldsymbol{A}$ and $\boldsymbol{B}$ are non-negative and $\mathbb{R}_+ = [0, \infty)$ denotes the non-negative real numbers.

Relation to other matrix factorizations techniques

NMF is related to many other techniques, such as vector quantization (VQ), principal component analysis (PCA), and independent component analysis (ICA), that can all be written as matrix factorizations on the form $\boldsymbol{X} \approx \boldsymbol{AB}$. The differences between these methods and NMF are due to different constraints placed on the factorizing matrices, $\boldsymbol{A}$ and $\boldsymbol{B}$: in VQ the columns of $\boldsymbol{B}$ are constrained to be unary vectors (all zero except one element equal to unity); in PCA columns of $\boldsymbol{A}$ and rows of $\boldsymbol{B}$ are constrained to be orthogonal; in ICA rows of $\boldsymbol{B}$ are maximally statistically independent; and in NMF all elements of $\boldsymbol{A}$ and $\boldsymbol{B}$ are non-negative. Several hybrid methods that combine these constraints have also been proposed, such as non-negative PCA [153, 172] and non-negative ICA [170, 171, 233].

Why non-negativity?

NMF is distinguished from other matrix factorization methods by the constraint that all elements in the factorizing matrices be non-negative. Many natural signals, such as pixel intensities, amplitude spectra, occurrence counts, and discrete probabilities, are naturally represented by non-negative numbers; thus, in the analysis of mixtures of such data, non-negativity of the individual components is a reasonable constraint. Also, non-negativity ensures that data is modeled as a purely additive combination of features, such that no cancellations can occur. This agrees with the intuitive idea of building the whole as the sum of its parts.

A brief note on history

Non-negative matrix factorization (NMF) was initially proposed by Paatero and Tapper [162][1]. Lee and Seung [128] later independently introduced NMF[2] as an unsupervised learning method used to model hand written digits. Subsequently they developed a simple multiplicative algorithm [126, 127] for computing the NMF based on two different divergence measures, and argued that NMF learns a "parts based" representation of data.

Review papers

There exist a few review papers on NMF that provide an overview of the related theory, algorithms, and applications. Berry et al. [22] describe the most fundamental NMF algorithms and discuss the use of auxiliary constraints used to impose prior knowledge on the problem. They illustrate applications of NMF with examples from text mining and spectral data analysis. Sra and Dhillon [209] provide a survey on NMF algorithms and applications with special focus on Bregman divergences. The survey includes an overview of application areas for NMF as well as a brief section on exact NMF. In a concise lecture note, Cichocki and Zdunek [46] present NMF and discuss cost functions, algorithms, and tensor extensions.

Basic computation

In its basic form, NMF can be computed as

$$\{\boldsymbol{A}, \boldsymbol{B}\} = \arg \min_{\boldsymbol{A},\boldsymbol{B} \geq \boldsymbol{0}} \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{B}), \tag{3.2}$$

where $\mathcal{D}$ is a cost function or divergence that measures the quality of approximation. That is, we find $\boldsymbol{A}$ and $\boldsymbol{B}$ that minimize the divergence between the data, $\boldsymbol{X}$, and the approximation, $\boldsymbol{AB}$. In general, NMF is not unique, and NMF algorithms thus usually find a local minimum of the divergence.

[1] Paatero and Tapper refer to the problem as *positive matrix factorization*.

[2] In their fist paper [128] on the subject, Lee and Seung refer to the problem as *conic coding*. In subsequent papers they use the term non-*negative matrix factorization*.

Maximum likelihood

NMF can be computed as a maximum likelihood estimate of $\boldsymbol{A}$ and $\boldsymbol{B}$ based on an assumption on the distributions of data. This assumption can be expressed in the likelihood function $p(\boldsymbol{X}|\boldsymbol{A},\boldsymbol{B})$. When we the choose the cost function, $\mathcal{D}$, to be the negative logarithm of the likelihood function,

$$\mathcal{D}_{\mathrm{ML}} = \mathcal{L}(\boldsymbol{X},\boldsymbol{AB}) = -\log\left[p(\boldsymbol{X}|\boldsymbol{A},\boldsymbol{B})\right], \tag{3.3}$$

we can compute the maximum likelihood (ML) estimate of $\boldsymbol{A}$ and $\boldsymbol{B}$ using (3.2).

Maximum a posteriori

In addition to the assumption on the distribution of data, we can also make assumptions on the distribution of the factors, expressed in terms of a prior distribution $p(\boldsymbol{A},\boldsymbol{B})$. Using Bayes rule, the posterior is given by $p(\boldsymbol{A},\boldsymbol{B}|\boldsymbol{X}) \propto p(\boldsymbol{X}|\boldsymbol{A},\boldsymbol{B})p(\boldsymbol{A},\boldsymbol{B})$, and by choosing the cost function

$$\begin{aligned}
\mathcal{D}_{\mathrm{MAP}} &= -\log\left[p(\boldsymbol{A},\boldsymbol{B}|\boldsymbol{X})\right] && (3.4)\\
&= \underbrace{-\log\left[p(\boldsymbol{X}|\boldsymbol{A},\boldsymbol{B})\right]}_{\mathrm{log-likelihood}} \underbrace{-\log\left[p(\boldsymbol{A},\boldsymbol{B})\right]}_{\mathrm{log-prior}} + c && (3.5)\\
&= \mathcal{L}(\boldsymbol{X},\boldsymbol{AB}) + \mathcal{P}(\boldsymbol{A},\boldsymbol{B}) + c, && (3.6)
\end{aligned}$$

where c is a constant, the maximum a posteriori (MAP) estimate of $\boldsymbol{A}$ and $\boldsymbol{B}$ can be computed using (3.2).

Ambiguities

Inherent to the NMF problem is a scale and permutation ambiguity; any solution is invariant to a permutation and scaling of the columns of $\boldsymbol{A}$ when the rows of $\boldsymbol{B}$ are permuted and inverse scaled correspondingly,

$$\boldsymbol{AB} = (\boldsymbol{APD})(\boldsymbol{D}^{-1}\boldsymbol{P}^{\top}\boldsymbol{B}) = \boldsymbol{A}^{*}\boldsymbol{B}^{*}, \tag{3.7}$$

where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{D}$ is any non-negative diagonal matrix. This means that in general we can only expect to recover $\boldsymbol{A}$ and $\boldsymbol{B}$ up to an arbitrary scaling and permutation; however, when computing a MAP estimate, the prior distribution of the factors may be used to resolve these ambiguities.

3.1 Applications of NMF

NMF has found widespread application in many different areas and has been used for both unsupervised and supervised learning. NMF and its generalizations and extensions has been used for such different purposes as dimensionality reduction, feature extraction, clustering, source separation and classification. In this section, a wide selection of the applications of NMF are reviewed. The review is organized according to application area.

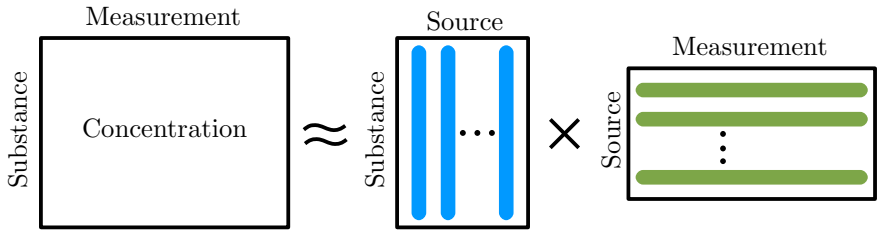

Figure 3.1: Illustration of NMF decomposition of environmetric data. Data is typically a series of measurements of concentrations of chemical substances, and the decomposition finds underlying explanatory sources.

3.1.1 Environmetrics and chemometrics

In environmetrics, NMF is often used to analyze series of chemical concentration measurements, to find underlying explanatory sources, as illustrated in Figure 3.1.

Anttila et al. [6] outline the use of NMF on environmental data, and analyze bulk wet deposition concentrations of chemical compounds. Lee et al. [129] apply NMF to the analysis of particle pollutants. Their data set consists of a series of measurements of concentrations of chemical species, and the factors found in the analysis correspond to different pollutant sources. Ramadan et al. [185] compare two NMF algorithms (PMF [162] and the multilinear engine [161]) for a similar problem. Kim et al. [111] show that the resolution of the method can be improved by incorporating auxiliary meteorological measurements.

NMF, and related methods, have been applied to a large number of curve resolution problems in chemometrics, where the purpose is to determine the spectra and concentration profiles of components in an unresolved mixture. NMF methods have been applied to data from liquid chromatography [73], reflectance spectroscopy [83], and Raman spectroscopy [137, 199]. Xie et al. [234] decompose pulsed gradient spin echo nuclear magnetic resonance data, using a three-way tensor extension to NMF, to resolve mixed chemical concentration-spectra in a solution.

3.1.2 Image processing

NMF has found several applications in image recognition and classification. In most applications images in a database are vectorized and the NMF is com-

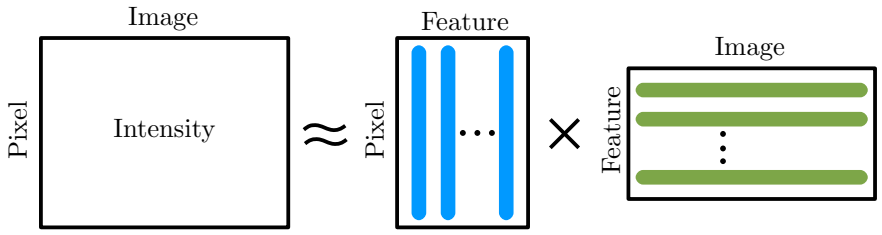

Figure 3.2: Illustration of NMF decomposition of images. Data is typically a set of vectorized images, and the decomposition finds a set of feature images.

puted on a matrix in which each column is an individual image as illustrated in Figure 3.2.

Lee and Seung [126] illustrates the use of NMF on a database of facial images and argue that the non-negative decomposition results in features that are "part based", i.e., the whole image is represented as the sum of its components. Mørup et al. [147] propose a 2-D shift invariant NMF method for extracting image features that are invariant to shifts in the plane.

Guillamet et al. [78, 82] use a weighted NMF [81] to classify patches of natural images. In later work, Guillamet and Vitrià [79, 80] apply NMF to the problem of recognizing faces under different conditions (expression, illumination, and occlusions.) Buciu and Pitas [32] compare PCA, NMF, and an extension called local NMF [138] for facial expression recognition. Liu and Zheng [143] show that image classification results can be improved by using a Riemannian distance metric or by orthogonalizing the bases learned by the NMF.

Hazan et al. [87] discuss the use of a tensor extension to NMF for sparse image coding. The method avoids vectorizing each image by representing a set of images as a three-dimensional tensor.

Cooper and Foote [52] apply NMF to generate video summaries, i.e., to find short passages of a video recording that are representative for the whole recording. Other applications of NMF to image data include image matting [109], i.e., to extract foreground objects and blend them into another scene; image unmixing [140]; and image fusion [243].

3.1.3 Text processing

NMF and probabilistic latent semantic analysis (PLSA) [93, 94] have found numerous applications in text analysis. The two methods are closely related, as

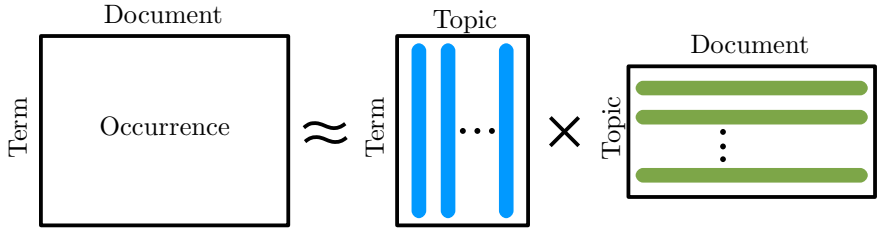

Figure 3.3: Illustration of NMF decomposition of text data. Data is typically a term-by-document matrix that contains the number of occurrences of each term in each document. The matrix if often sparse, since most terms only occur in few documents. The decomposition finds sets of related terms, corresponding to topics in the set of documents.

it has been shown [74], that the PLSA algorithm solves the NMF problem with a Kullback-Leibler divergence. Most often NMF is used to analyze a term-by-document occurrence matrix to find topics as illustrated in Figure 3.3.

Lee and Seung [126] analyze a corpus of documents summarized by a term-by-document occurrence matrix, and show that the factors found by NMF correspond to semantic features (topics). Novak and Mammone [149, 150] use NMF to construct a language model for automated speech recognition. They show that this leads to better results in terms of perplexity, in comparison with latent semantic analysis (LSA). Tsuge et al. [216] show that the precision in a document query task is significantly improved when the dimensionality of a term-by-document matrix is reduced using NMF, and distances between queries and documents are measured in the reduced-dimensional space.

For the problem of clustering a corpus of documents in groups of semantically related documents, Xu et al. [235] show that NMF outperforms LSA and spectral clustering. Shahnaz et al. [202] propose a regularized NMF method [167] that further improve results. Berry and Brown [21] apply this method to the Enron email data set, and suggest that the method could be used for automatic email surveillance.

3.1.4 Audio processing

NMF has a multitude of applications in audio processing, including feature extraction, music transcription, sound classification, and source separation. Most NMF decompositions of audio data are computed in a time-by-frequency representation as illustrated in Figure 3.4.

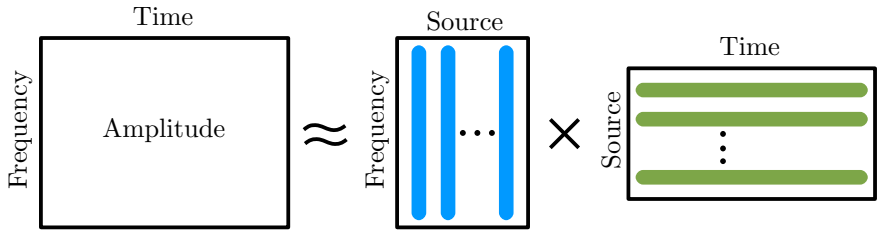

Figure 3.4: Illustration of NMF decomposition of audio. Data is typically a time-by-frequency matrix, such as the magnitude short time Fourier transform. The decomposition finds a set of time-varying sources with constant spectrum.

Sha and Saul [201] use NMF to estimate multiple fundamental frequencies of simultaneous acoustic sources, based on an instantaneous frequency estimation [2, 27, 71] preprocessing step. They report that the method successfully estimates the fundamental frequency of two overlapping speech sources. On a similar problem, Raczyński et al. [174] propose a harmonically constrained NMF method that is reported to give improved results on a note detection task compared with traditional NMF.

Smaragdis and Brown [204] use a sparse NMF approach for transcription of polyphonic music, by learning spectral profiles for each note. A similar system is proposed by Abdallah and Plumbley [1] who also provide a rigorous probabilistic foundation.

Cho et al. [39, 40] use NMF for learning spectral features for an audio classification task and demonstrate an improvement in recognition accuracy compared with features based on independent component analysis.

Several authors propose to use NMF for audio source separation. Wang and Plumbley [225] use NMF to decompose an audio signal into components, that are manually grouped to form individual audio sources. Similarly, Helén and Virtanen [90] use NMF to separate polyphonic music into components, and for each component they extract a set of features and use a support vector machine (SVM) to classify the component as either harmonic or drum.

Smaragdis [208] and Virtanen [222] independently introduce convolutive extensions of NMF, in which each component is allowed to have a time-varying spectrum. This enables the NMF basis functions to capture transients. Schmidt and Mørup [A] propose a 2-D convolutive NMF method for blind separation of music instruments, that extends the convolutive NMF by introducing an invariance to shifts on a logarithmic frequency axis, corresponding to a change of pitch.

For the problem of separating multiple simultaneous speakers, Raj et al. [183, 184] introduce a probabilistically motivated NMF model-based on a mixture of multinomial distributions over frequency bins. Schmidt and Olsson [B] propose a method where an over-complete basis is computed for a set of speakers using sparse NMF. In later work [D] they improve results using linear regression in the sparse NMF feature space.

3.1.5 Bioinformatics

Bioinformatics is another large application area for NMF. Several authors use NMF to analyze micro-array gene-expression data, in many papers with the purpose of distinguishing between different types of cancer [30, 69, 70, 72, 113, 187]. Wang et al. [227] present an NMF method that utilizes the uncertainty estimates for each data point that are often available in micro-array data.

NMF is applied to electroencephalogram (EEG) signal classification by several authors [130, 236]. Chen et al. [38] introduce a constrained NMF method with temporal smoothness and spatial decorrelation for detection of Alzheimer's disease using EEG recordings. A tensor extension to NMF, that directly model multichannel EEG recordings, is proposed by Lee et al. [131].

Sajda et al. [199, 200] analyze chemical shift imaging data of the human brain, and use NMF to distinguish between brain and muscle tissue. For the same data set, Schmidt and Laurberg [E] introduce an NMF method Gaussian process priors and show that this leads to better separation.

Lee et al. [132, 133] apply NMF to myocardial positron emission tomography (PET) images and find a basis that corresponds to major cardiac components. They report that results are similar to those obtained using factor analysis. On a similar data set, Ahn et al. [3] use a multilayer NMF method to obtain a hierarchical decomposition.

Hoyer [98] suggests modeling the processing in the the early visual system (V1) using a sparse NMF method. He shows that an analysis of a database of natural images results in features that resemble the simple cell receptive fields in V1.

3.1.6 Other applications

NMF has also been used in a number of other applications, a few of which are mentioned here. Several authors use NMF for analyzing astronometric data, including molecular emission spectra [108] and spectral reflectance [166, 168].

Buchsbaum and Bloch [31] analyze color spectra and observe that the components in the NMF correspond to established color naming categories. Ramanath et al. [186] compare perceptual color spaces with color spaced obtained using dimensionality reduction techniques such as PCA, ICA, and NMF. Young et al. [237] use NMF to find characteristic flavor profiles in Scotch single malt whiskeys. Hu et al. [99] propose to use NMF to find ratio rules, i.e., events that occur at characteristic fixed ratios, in a basketball statistics data set.

3.2 Generalizations and extensions of NMF

Many different generalizations and extensions to NMF have been proposed in the literature. Some generalizations fit directly in the NMF framework, and deal with finding non-negative factorization with specific properties. This includes different NMF cost functions, some of which arise from assumptions of the distribution of the data, and different methods for finding factors with desired characteristics such as sparsity, orthogonality, smoothness, symmetries, and invariances. Other methods extend the NMF framework, for example non-negative factorization of tensors (multidimensional arrays); convolutive models, hierarchical/multilayer models, and models which relax the non-negativity constraints. In this section, a selection of generalizations and extensions of NMF are reviewed.

3.2.1 Divergence measures

A wide range of different cost functions have been proposed for NMF in the literature, most often expressed in terms of a divergence measure, $\mathcal{L}(\boldsymbol{X}, \boldsymbol{AB})$. In general, these divergences are not symmetric, and for some of these asymmetric divergence measures there also exists a useful dual divergence, $\mathcal{L}(\boldsymbol{AB}, \boldsymbol{X})$. Computing the NMF by minimizing a divergence measure in many cases corresponds to computing the maximum likelihood estimate under certain assumptions on the distribution of the data.

The arguably most simple and most widely used cost function for the NMF problem is the least squares (LS) cost

$$\mathcal{L}_{\text{LS}} = \sum_{i,j} (\boldsymbol{X} - \boldsymbol{AB})_{i,j}^2, \tag{3.8}$$

that corresponds to the assumption that the residual is i.i.d. Gaussian distributed.

Lee and Seung [126] introduce a cost function

$$\mathcal{L}_{\mathrm{P}} = \sum_{i,j} (\boldsymbol{AB})_{i,j} - \boldsymbol{X}_{i,j} \log(\boldsymbol{AB})_{i,j}, \tag{3.9}$$

that can be derived on the assumption that $\boldsymbol{X}_{i,j}$ follows a Poisson distribution with mean $(\boldsymbol{AB})_{i,j}$.

The Poisson cost function can also be seen as a special case of the generalized Kullback-Leibler (KL) divergence

$$\mathcal{L}_{\mathrm{KL}} = \sum_{i,j} \boldsymbol{X}_{i,j} \log \frac{\boldsymbol{X}_{i,j}}{(\boldsymbol{AB})_{i,j}} - \boldsymbol{X}_{i,j} + (\boldsymbol{AB})_{i,j}, \tag{3.10}$$

that measures the relative entropy between the data and the approximate factorization, if $\boldsymbol{X}$ can be considered as an unnormalized discrete probability distribution.

Dhillon and Sra [54] propose to use the Bregman divergence for NMF. The Bregman divergence generalizes the least squares and the generalized Kullback-Leibler divergences. For any continuously-differentiable strictly convex function, ψ, there exists a Bregman divergence defined as

$$\mathcal{L}_{\mathrm{B}} = \sum_{i,j} \psi(\boldsymbol{X}_{i,j}) - \psi((\boldsymbol{AB})_{i,j}) - \nabla\psi\left((\boldsymbol{AB})_{i,j}\right)(\boldsymbol{X} - \boldsymbol{AB})_{i,j}, \tag{3.11}$$

which corresponds to (3.8) for $\psi(x) = \frac{1}{2}x^2$ and to (3.10) for $\psi(x) = x \log x - x$. It can be shown [15] that there exists a bijection between Bregman divergences and exponential family distributions. This means that if we assume the residual follows some specific exponential family distribution, there is a corresponding Bregman divergence that can be used to compute the maximum likelihood NMF.

Another divergence measure that generalizes the least squares and the Kullback-Leibler divergences is proposed by Kompass [120]

$$\mathcal{L}_{\mathrm{K}} = \sum_{i,j} \boldsymbol{X}_{i,j} \frac{\boldsymbol{X}_{i,j}^{\alpha} - (\boldsymbol{AB})_{i,j}^{\alpha}}{\alpha} - (\boldsymbol{AB})_{i,j}^{\alpha}(\boldsymbol{AB} - \boldsymbol{X})_{i,j}, \tag{3.12}$$

and corresponds to least squares for $\alpha = 1$ and generalized KL in the limit $\alpha \to 0$.

Cichocki et al. [49] discuss the use of the Csiszár divergence in NMF

$$\mathcal{L}_{\mathrm{C}} = \sum_{i,j} \boldsymbol{X}_{i,j}\ \varphi\left(\frac{(\boldsymbol{AB})_{i,j}}{\boldsymbol{X}_{i,j}}\right), \tag{3.13}$$

where φ is a strictly convex function with $\varphi(1) = 0$. This family of divergences generalizes a large number of other known divergences, including the KL, $\varphi(x) = \log x + \frac{1}{x} - 1$; the dual KL divergence, $\varphi(x) = x \log x - x + 1$; and Amari's alpha divergence, $\varphi(x) = \frac{x^{\beta-1}-1}{\beta(\beta-1)} + \frac{x-1}{\beta}$.

Weighted NMF

NMF was initially introduced [162] as a weighted least squares estimate, i.e., with the cost function

$$\mathcal{L}_{\text{WLS}} = \sum_{i,j} \boldsymbol{W}_{i,j} (\boldsymbol{X} - \boldsymbol{AB})_{i,j}^2, \tag{3.14}$$

where $\boldsymbol{W}$ is a matrix of weights. When the standard deviations, $\boldsymbol{\sigma}_{i,j}$, of the data points are known, the weights can be selected as $\boldsymbol{W}_{i,j} = 1/\boldsymbol{\sigma}_{i,j}^2$. This corresponds to the assumption that the elements of the residual are independent Gaussian distributed with zero mean and variance $\boldsymbol{\sigma}_{i,j}^2$.

Guillamet et al. [81, 82] introduce a column-wise weighted version of the Poisson cost function in (3.9)

$$\mathcal{L}_{\text{WP}} = \sum_{i,j} \boldsymbol{w}_j (\boldsymbol{AB})_{i,j} - \boldsymbol{w}_j \boldsymbol{X}_{i,j} \log \left(\boldsymbol{w}_j (\boldsymbol{AB})_{i,j} \right), \tag{3.15}$$

where $\boldsymbol{w}$ is a vector of weights. When columns of $\boldsymbol{X}$ are considered as training vectors, the authors suggest giving more weight to vector that have a low probability of appearing in the training set, but a high probability of occurring in the assumed underlying distribution, to counter sample selection bias.

3.2.2 Distribution of the factors

In standard NMF methods, the only assumptions made about the factors in the model is that they are non-negative. In probabilistic terms, we can think of this as an non-informative improper prior over the non-negative real numbers. In the literature, several methods have been proposed for finding NMF decompositions where the factors have other properties of interest, such as sparseness or smoothness. Often, the decompositions are computed by minimizing a cost function augmented by penalty or regularization terms that account for these constraints on the factors. Many of these methods can also be interpreted as maximum a posteriori estimates of the factorization with specific prior distributions over the factors.

Sparsity

Arguably the most important extension of NMF in terms of alternative distributions of the factors is sparse NMF, where the objective is to find a factorization,

$\boldsymbol{X} \approx \boldsymbol{AB}$, where $\boldsymbol{B}$ is sparse, i.e., most of its elements are zero. Denoted non-negative sparse coding (NNSC), a method for sparse NMF is introduced by Hoyer [96] based on minimizing a penalized least squares cost function. The proposed penalty term is

$$\mathcal{P}_{\text{NNSC}} = \beta \sum_{n,j} f(\boldsymbol{B}_{n,j}), \tag{3.16}$$

where β is a parameter that controls the trade-off between sparseness and reconstruction error and f is a function that measures sparseness. A typical choice [96] is $f(x) = |x|$, which is also known as an L_1 norm regularization. This corresponds to the assumption that the elements in $\boldsymbol{B}$ are i.i.d. one-sided exponential. Hoyer [96] points out an important problem with this cost function when f is an increasing function: because of the scale ambiguity inherent to the NMF problem, the second term in the cost function can be trivially minimized by letting $\boldsymbol{A}$ increase and $\boldsymbol{B}$ decrease correspondingly. This is easily remedied, however, by imposing a hard constraint on the scale of either $\boldsymbol{A}$ or $\boldsymbol{B}$, for example by forcing the norm of the columns of $\boldsymbol{A}$ to unity.

In a later paper [97], Hoyer introduces another sparsity measure, defined for a vector $\boldsymbol{x}$,

$$s(\boldsymbol{x}) = \frac{\sqrt{n} - \frac{\sum_i |\boldsymbol{x}_i|}{\sqrt{\sum_i \boldsymbol{x}_i^2}}}{\sqrt{n} - 1}, \tag{3.17}$$

where n is the dimensionality of $\boldsymbol{x}$. Based on the relation between the L_1 and L_2 norm, the measure is equal to zero when all elements of $\boldsymbol{x}$ are equal and it is equal to one when only a single element of $\boldsymbol{x}$ is non-zero. Hoyer proposes to minimize the least squares cost function under the constraints, $s(\boldsymbol{A}_n) = S_A$, $\forall n$, and $s(\boldsymbol{B}_n) = S_B$, $\forall n$, where $\boldsymbol{A}_n$ is the nth column of $\boldsymbol{A}$, $\boldsymbol{B}_n$ is the nth row of $\boldsymbol{B}$, and S_A, S_B are the desired sparsity of the factors.

Stadlthanner et al. [213] extend the sparse NMF to allow different sparsity constraints for each feature (columns of $\boldsymbol{A}$ and rows of $\boldsymbol{B}$), which is non-trivial because of the permutation ambiguity in the NMF problem. The authors present an algorithm in which the factors are adaptively permuted according to their sparsity measure.

A different approach, denoted non-smooth NMF, is taken by Pascual-Montano et al. [165], who propose to solve a modified NMF problem

$$\boldsymbol{X} \approx \boldsymbol{ASB}, \tag{3.18}$$

where $\boldsymbol{S} \in \mathbb{R}_+^{n \times n}$ is a smoothing matrix

$$\boldsymbol{S} = (1 - \theta)\boldsymbol{I} + \theta \boldsymbol{1}. \tag{3.19}$$

Here, $\boldsymbol{I}$ is the identity matrix, $\boldsymbol{1}$ is a matrix with all elements equal to one, and $\theta \in [0, 1]$ is used to control the degree of sparsity. Since the introduction of $\boldsymbol{S}$ smooths the factorization, the resulting factors will be more sparse or non-smooth to oppose the smoothing.

Orthogonality

Ding et al. [58] discuss NMF with orthogonality constraints

$$\boldsymbol{X} \approx \boldsymbol{AB}, \quad \text{s.t. } \boldsymbol{A}, \boldsymbol{B} \geq \boldsymbol{0},\ \boldsymbol{BB}^\top = \boldsymbol{I}, \tag{3.20}$$

and show that it is equivalent to k-means clustering. Intuitively, since $\boldsymbol{B}$ is non-negative, the orthogonality constraint implies that only one element in each column of $\boldsymbol{B}$ can be non-zero and this leads to a clustering of the data. The authors further discuss a bi-orthogonal tri-factorization

$$\boldsymbol{X} \approx \boldsymbol{ASB}, \quad \text{s.t. } \boldsymbol{A}, \boldsymbol{S}, \boldsymbol{B} \geq \boldsymbol{0},\ \boldsymbol{A}^\top\boldsymbol{A} = \boldsymbol{BB}^\top = \boldsymbol{I}, \tag{3.21}$$

and show its relation to kernel k-means clustering for a specific kernel function.

As a part of their local NMF method, Li et al. [65, 138] propose a method for finding factors that are not strictly orthogonal but are optimized for maximum orthogonality. In addition to orthogonality, the local NMF method also maximizes sparseness and expressiveness; however, here we only discuss the proposed approach to orthogonalization. The authors propose a cost function penalized by

$$\mathcal{P}_\mathrm{O} = \alpha \sum_{i \neq n} (\boldsymbol{A}^\top\boldsymbol{A})_{i,n}, \tag{3.22}$$

where α is a parameter that makes a trade-off between orthogonality and reconstruction error. Similar to the non-negative sparse coding penalty term in (3.16), this can be trivially minimized by decreasing $\boldsymbol{A}$ and increasing $\boldsymbol{B}$ correspondingly, so a constraint on the scale of either $\boldsymbol{A}$ or $\boldsymbol{B}$ must be enforced.

Because of the permutation ambiguity, the factors computed by most NMF methods occur in arbitrary order. Li et al. [137] propose a NMF method where an orthogonality constraint is enforced between $\boldsymbol{A}$ and a fixed reference. This approach does not lead to factors that are orthogonal to each other, but it provides a means for steering the solution of the NMF problem away from a specified reference.

Smoothness

In many applications the data matrix, $\boldsymbol{X}$, consists of consecutive samples from some slowly varying process and thus $\boldsymbol{X}$ is smooth in one or possibly both directions. When this is the case, it is natural to enforce a continuity or smoothness constraint in the NMF decomposition.

For this aim, Virtanen [220] proposes to minimize the absolute value of the difference between the elements in the rows of $\boldsymbol{B}$

$$\mathcal{P}_{\mathrm{S1}} = \alpha \sum_{n,j} |\boldsymbol{B}_{n,j-1} - \boldsymbol{B}_{n,j}|. \tag{3.23}$$

Again, this penalty term requires a constraint on the scale of $\boldsymbol{A}$ or $\boldsymbol{B}$ to avoid trivial minimization. In later work [221], Virtanen proposes a penalty based on the squared difference

$$\mathcal{P}_{\mathrm{S2}} = \alpha \sum_{n,j} \frac{(\boldsymbol{B}_{n,j-1} - \boldsymbol{B}_{n,j})^2}{\frac{1}{N}\sum_{n} \boldsymbol{B}_{n,j}^2}, \tag{3.24}$$

where the penalty for each row of $\boldsymbol{B}$ is normalized by its mean square, which makes the expression invariant to the scale of $\boldsymbol{B}$. Virtanen argues [220] that measuring smoothness using the absolute value of differences preserves rapid changes better than using the squared differences. For example, for a change from zero to a constant level any sequence of non-decreasing steps will have an equal sum of absolute value differences, whereas the squared difference penalty favors many small steps of equal size. Both $\mathcal{P}_{\mathrm{S1}}$ and $\mathcal{P}_{\mathrm{S2}}$ reach a (trivial) global minimum when the rows of $\boldsymbol{B}$ are constant, and the parameter α is used to make a trade-off between smoothness and data fit.

Chen and Cichocki [37] propose a different measure of smoothness based on the squared difference between $\boldsymbol{B}$ and a matrix $\bar{\boldsymbol{B}}$, in which each row is a low-pass filtered version of the corresponding row of $\boldsymbol{B}$

$$\mathcal{P}_{\mathrm{LP}} = \alpha \sum_{n,j} (\boldsymbol{B} - \bar{\boldsymbol{B}})_{n,j}^2. \tag{3.25}$$

For the low-pass filter, the authors use an exponentially weighted moving average, and show that this can be implemented efficiently.

Discriminative factors

Often, NMF is used to transform a data matrix into a set of features that are used to perform some desired task. In its basic form, NMF finds the set of non-negative features that best fit the data, according to some cost function. When the features are to be used subsequently in a classification task, however, the objective is to find features that discriminate well between different classes.

For this means, Wang et al. [228] present a supervised discriminative NMF method that is based on a penalized KL cost function. They propose a penalty term inspired by Fisher discriminant analysis, that minimize within-class scatter

while maximizing between-class scatter,

$$\mathcal{P}_{\mathrm{F}} = \frac{\alpha}{C} \sum_{c,n} \left(\frac{1}{|\xi_c|} \sum_{j \in \xi_c} \left(\boldsymbol{B}_{n,j} - \boldsymbol{b}_n^{(c)} \right)^2 - \frac{1}{C-1} \sum_{c'} \left(\boldsymbol{b}_n^{(c)} - \boldsymbol{b}_n^{(c')} \right)^2 \right), \quad (3.26)$$

where C is the number of classes, ξ_c denotes the set of indices j that belong to class c, and

$$\boldsymbol{b}_n^{(c)} = \frac{1}{|\xi_c|} \sum_{j \in \xi_c} \boldsymbol{B}_{n,j} \quad (3.27)$$

is the mean of the columns of $\boldsymbol{B}$ that belong to class c. Wang et al. demonstrate that the discriminative NMF improves the performance compared to regular NMF on a face recognition task.

Another approach to discriminative NMF is taken by Kim and Park [112], who propose a method where a NMF is computed separately on each class. The combined set of basis functions is then used to compute features. The authors demonstrate improved performance on several classification tasks, compared with a nearest neighbor classifier based on a NMF of data from all classes combined.

Gaussian process priors

Schmidt and Laurberg [E] present a general method for including prior knowledge in NMF based on Gaussian process priors. In this approach, the non-negative factors $\boldsymbol{A}$ and $\boldsymbol{B}$ are linked by strictly increasing functions, f_A and f_B, to underlying Gaussian processes, $\boldsymbol{a}$ and $\boldsymbol{b}$,

$$\boldsymbol{a} = f_A\left[\mathrm{vec}(\boldsymbol{A})\right], \ \boldsymbol{b} = f_B\left[\mathrm{vec}(\boldsymbol{B})\right]. \quad (3.28)$$

The Gaussian processes, $\boldsymbol{a}$ and $\boldsymbol{b}$, are fully specified by their covariance functions, and the link and covariance functions are used to control the properties of the factors in the NMF, such as sparsity, smoothness, and symmetries.

3.2.3 Structured factors

Convolutive NMF

Smaragdis [208] and Virtanen [222] independently propose an extension of the NMF model where the matrix product $\boldsymbol{AB}$ is extended to a discrete convolution

$$\boldsymbol{X}_{i,j} \approx \sum_{n,k} \boldsymbol{A}_{i,n,k} \boldsymbol{B}_{n,j-k}. \quad (3.29)$$

Smaragdis applies [205, 207, 208] the method to decomposition of audio spectrograms, where $\boldsymbol{X}$ is a time-by-frequency matrix, and argues that each component in the NMF corresponds to an auditory object. The convolutive extension allows these objects to have a temporal structure.

The convolutive NMF can formulated as a matrix product like the usual NMF problem, $\boldsymbol{X} \approx \boldsymbol{A}\bar{\boldsymbol{B}}$, where $\boldsymbol{A} \in \mathbb{R}_+^{I \times NK}$ is a general non-negative matrix, and $\bar{\boldsymbol{B}} \in \mathbb{R}_+^{NK \times J}$ has the following structure

$$\bar{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{B}\boldsymbol{J} \\ \boldsymbol{B}\boldsymbol{J}^2 \\ \vdots \\ \boldsymbol{B}\boldsymbol{J}^{K-1} \end{bmatrix}, \tag{3.30}$$

where $\boldsymbol{J}$ is a square matrix with ones on the first lower sub-diagonal and zeros elsewhere, such that post-multiplication by $\boldsymbol{J}$ corresponds to "shifting" the columns of $\boldsymbol{B}$ one position to the left. This formulation makes it clear that the convolutive NMF model is a specific structured NMF in which some elements of the $\boldsymbol{B}$ matrix are constrained to be equal, analogous to weight-sharing in artificial neural networks. Virtanen [222] and O'Grady and Pearlmutter [152] further extend the convolutive NMF by adding sparsity constraints.

Schmidt and Mørup [A] extend the convolutive NMF model to a two-dimensional convolution

$$\boldsymbol{X}_{i,j} \approx \sum_{n,k,l} \boldsymbol{A}_{i-l,n,k} \boldsymbol{B}_{n,j-k,l}, \tag{3.31}$$

which can also be formulated as a structured NMF, $\boldsymbol{X} \approx \tilde{\boldsymbol{A}}\tilde{\boldsymbol{B}}$, where both $\tilde{\boldsymbol{A}} \in \mathbb{R}_+^{I \times NKL}$ and $\tilde{\boldsymbol{B}} \in \mathbb{R}_+^{NKL \times J}$ are structured matrices

$$\tilde{\boldsymbol{A}} = \left[\boldsymbol{A}^{(1)} \cdots \boldsymbol{A}^{(K)} \cdots {\boldsymbol{J}^{L-1}}^\top \boldsymbol{A}^{(1)} \cdots {\boldsymbol{J}^{L-1}}^\top \boldsymbol{A}^{(K)} \right], \tag{3.32}$$

$$\tilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B}^{(1)} \\ \vdots \\ \boldsymbol{B}^{(1)}\boldsymbol{J}^{K-1} \\ \vdots \\ \boldsymbol{B}^{(L)} \\ \vdots \\ \boldsymbol{B}^{(L)}\boldsymbol{J}^{K-1} \end{bmatrix}. \tag{3.33}$$

All the parameters in the model are compactly represented by the two sets of matrices $\{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(K)}\} \in \mathbb{R}_+^{I \times N}$ and $\{\boldsymbol{B}^{(1)}, \ldots, \boldsymbol{B}^{(L)}\} \in \mathbb{R}_+^{N \times J}$. The authors

apply the two-dimensional convolutive NMF to single-channel separation of musical instruments, where $\boldsymbol{X}$ is a time-by-frequency matrix. The convolutions in time and frequency correspond to models of temporal dynamics and pitch-shift. Mørup et al. [147] and FitzGerald et al. [66] further extend the 2-D convolutive NMF to decompositions of tensors, and FitzGerald et al. [67] extend the model by harmonicity constraints. NMF and its convolutive variants are illustrated in Figure 3.5.

Transformation invariant NMF

Wersing et al. [62, 231] propose a (sparse) transformation invariant NMF model

$$\boldsymbol{X}_{i,j} \approx \sum_{n,k} \boldsymbol{A}_{i,n,k} \left[\boldsymbol{T}^{(k)}(\boldsymbol{B}) \right]_{n,j}, \tag{3.34}$$

where $\boldsymbol{T}^{(k)}$, $k \in \{1, \ldots, K\}$, is a fixed set of transformations. Comparison with (3.29) shows that convolutive NMF is a special case of (3.34) where the set of transformations correspond to shifting columns in the $\boldsymbol{B}$ matrix, $\boldsymbol{T}^{(k)}(\boldsymbol{B}) = \boldsymbol{B}\boldsymbol{J}^k$. The authors apply the transformation invariant NMF to the problem of finding a translation invariant basis for a set of images, and discuss the possibility of extending the set of transformations to include scaling, rotation, and other more advanced transformations.

Similar to the convolutive models, transformation invariant NMF can formulated as a matrix product $\boldsymbol{X} \approx \boldsymbol{A}\hat{\boldsymbol{B}}$, where $\boldsymbol{A} \in \mathbb{R}_+^{I \times NK}$ is a general non-negative matrix, and $\hat{\boldsymbol{B}} \in \mathbb{R}_+^{NK \times J}$ has the following structure

$$\hat{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{T}^{(1)}(\boldsymbol{B}) \\ \boldsymbol{T}^{(2)}(\boldsymbol{B}) \\ \vdots \\ \boldsymbol{T}^{(K)}(\boldsymbol{B}) \end{bmatrix}. \tag{3.35}$$

Generalized structured factors

The convolutive and transformation invariant NMF models can more generally be written as

$$\boldsymbol{X} \approx \boldsymbol{A}(\boldsymbol{a})\boldsymbol{B}(\boldsymbol{b}), \tag{3.36}$$

where the non-negative matrices $\boldsymbol{A} \in \mathbb{R}_+^{I \times N}$ and $\boldsymbol{B} \in \mathbb{R}_+^{N \times J}$ are determined by two vectors of parameters, $\boldsymbol{a}$ and $\boldsymbol{b}$. These parameters, in general, need not be non-negative; however, $\boldsymbol{A}(\boldsymbol{a})$ and $\boldsymbol{B}(\boldsymbol{b})$ must be non-negative for any valid choice of $\boldsymbol{a}$ and $\boldsymbol{b}$. The Gaussian process prior framework proposed by Schmidt and Laurberg [E] can be seen as as a special case of (3.36) where $\boldsymbol{a}$ and $\boldsymbol{b}$ are modeled as Gaussian processes.

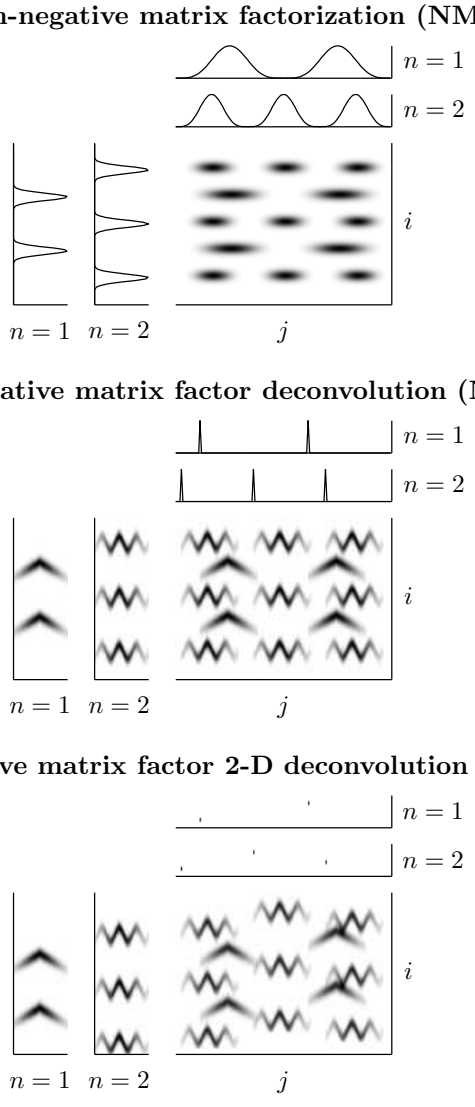

Figure 3.5: Illustration of non-negative matrix factorization (NMF), Non-negative matrix factor deconvolution (NMFD), and non-negative matrix factor 2-D deconvolution (NMF2D), each with two components. In each figure, $\boldsymbol{X}$ is shown at the bottom right, the two components of $\boldsymbol{A}$ are to the left, and the two components of $\boldsymbol{B}$ are at the top.

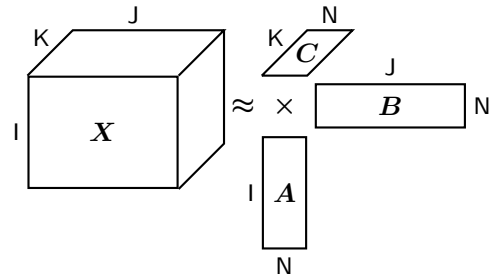

Figure 3.6: Illustration of the three-dimensional PARAFAC model.

3.2.4 Tensor extensions

NMF is in its basic formulation a non-negative bilinear decomposition of a two-dimensional array, but it can be extended to factorization of tensors (multidimensional arrays) with any number of modes, for example tri-linear decompositions of three-dimensional arrays. The idea of factorizing multidimensional arrays dates back to Hitchcock [91, 92] (for a review, see e.g. Kolda and Bader [119]), and several approaches to non-negative tensor factorization have been proposed in the literature. In this section, the discussion is limited to three-way factorizations, but non-negative factorizations of higher order can be formulated as well.

PARAFAC model

Paatero [159, 160, 161] extends NMF to a three-way factorization based on the parallel factor analysis (PARAFAC) [35, 85] model, which is also known as the canonical decomposition (candecomb) model. For three-dimensional data, the non-negative PARAFAC model can be written as

$$\boldsymbol{X}_{i,j,k} \approx \sum_{n} \boldsymbol{A}_{i,n} \boldsymbol{B}_{j,n} \boldsymbol{C}_{k,n}, \tag{3.37}$$

where $\boldsymbol{X} \in \mathbb{R}^{I \times J \times K}$ is a three-dimensional tensor. This decomposition can be seen as a sum of outer products of the columns of the factor matrices, $\boldsymbol{A} \in \mathbb{R}_{+}^{I \times N}$, $\boldsymbol{B} \in \mathbb{R}_{+}^{J \times N}$, and $\boldsymbol{C} \in \mathbb{R}_{+}^{K \times N}$, and is as such arguably the most straightforward tensor extension of NMF. Welling and Weber [230] generalize the model to tensors of arbitrary dimensionality. The non-negative tree-way PARAFAC model is illustrated in Figure 3.6.

Another non-negative three-way factorization, based on the PARAFAC2[3] [86] model, is discussed by Cichocki et al. [50, 51]. The three-way PARAFAC2

[3]Cichocki et al. denote this *non-negative tensor factorization 2 (NTF2)*

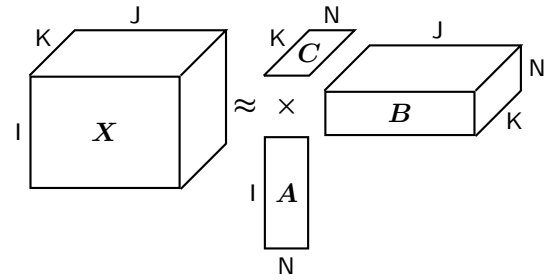

Figure 3.7: Illustration of the three-dimensional PARAFAC2 model.

decomposition can be written as

$$\boldsymbol{X}_{i,j,k} \approx \sum_{n} \boldsymbol{A}_{i,n} \boldsymbol{B}_{j,k,n} \boldsymbol{C}_{k,n}, \tag{3.38}$$

and differs from the PARAFAC model in that the factor $\boldsymbol{B} \in \mathbb{R}^{J \times K \times N}$ is itself a three-way tensor. The non-negative three-way PARAFAC2 model is illustrated in Figure 3.7.

Tucker model

Kim et al. [116, 118] discuss a non-negative tensor decomposition based on the the Tucker [217] model

$$\boldsymbol{X}_{i,j,k} \approx \sum_{l,m,n} \boldsymbol{A}_{i,l} \boldsymbol{B}_{j,m} \boldsymbol{C}_{k,n} \boldsymbol{G}_{l,m,n}. \tag{3.39}$$

Here, the tensor $\boldsymbol{X} \in \mathbb{R}^{I \times J \times K}$ is decomposed into three non-negative matrices of different dimensionality, $\boldsymbol{A} \in \mathbb{R}_{+}^{I \times L}$, $\boldsymbol{B} \in \mathbb{R}_{+}^{J \times M}$, and $\boldsymbol{C} \in \mathbb{R}_{+}^{K \times N}$ that are coupled together by a non-negative core matrix, $\boldsymbol{G} \in \mathbb{R}_{+}^{L \times M \times N}$. The non-negative three-way Tucker model is illustrated in Figure 3.8.

3.2.5 Other extensions and relations

Symmetric NMF and clustering

When the data matrix $\boldsymbol{X}$ is symmetric, the NMF may posses certain interesting properties. Catral et al. [36] discuss the conditions i) under which the approximating factors are equal, $\boldsymbol{A} = \boldsymbol{B}^{\top}$, and ii) under which the NMF of a symmetric $\boldsymbol{X}$ yields a symmetric approximation, $\boldsymbol{AB} = (\boldsymbol{AB})^{\top}$.

Ding et al. [55] presents an algorithm for computing a symmetric NMF decomposition, $\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{A}^{\top}$, and shows that this corresponds to special cases of both kernel k-means clustering and spectral clustering.

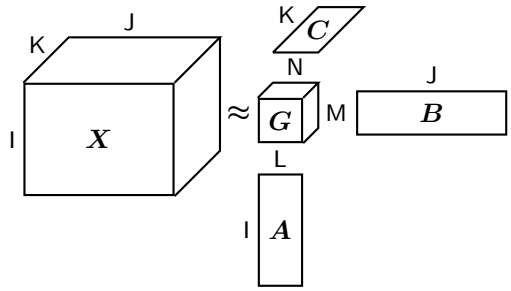

Figure 3.8: Illustration of the three-dimensional Tucker model.

Probabilistic latent semantic analysis (PLSA) [93, 94] is an unsupervised learning method based on a statistical latent variable model for co-occurrence data that has been applied to text analysis tasks such as document clustering. Xu et al. [235] presents a least squares NMF clustering method for document clustering, that is closely related to the PLSA method. Gaussier and Goutte [74] and Ding et al. [57] show that PLSA is equivalent to NMF with the KL-divergence, in the sense that the two methods minimize the same cost function.

Several matrix factorization methods can be used for clustering. Li and Ding [139] provide an overview over different matrix factorizations including NMF and several NMF variants, and compare the methods in a clustering context.

Many unsupervised clustering algorithms, including most methods based on NMF, are sensitive to initial conditions, and the resulting clusters obtained on a dataset may vary between runs of the algorithm. Badea [12] presents a meta-clustering algorithm based on NMF in which a dataset is clustered several times resulting in a set of clusters that are subsequently clustered to yield meta-clusters. The authors show that the meta-clustering method is substantially improved when using a soft NMF clustering method compared with a hard k-means clustering.

Hierarchical models

Cichocki and Zdunek [43, 44] present a hierarchical approach to NMF where the data is first modeled by a standard NMF, $\boldsymbol{X} \approx \boldsymbol{A}^{(1)}\boldsymbol{B}^{(1)}$, where $\boldsymbol{A}^{(1)} \in \mathbb{R}^{I\times N}$. In a subsequent step, the matrix $\boldsymbol{B}^{(1)} \in \mathbb{R}^{N\times J}$ is modeled by NMF as $\boldsymbol{B}^{(1)} \approx \boldsymbol{A}^{(2)}\boldsymbol{B}^{(2)}$, where $\boldsymbol{A}^{(2)} \in \mathbb{R}^{N\times N}$ and the process is continued by computing $\boldsymbol{B}^{(l)} \approx \boldsymbol{A}^{(l+1)}\boldsymbol{B}^{(l+1)}$ for L iterations, yielding a hierarchical NMF model

$$\boldsymbol{X} \approx \boldsymbol{A}^{(1)}\boldsymbol{A}^{(2)}\cdots\boldsymbol{A}^{(L)}\boldsymbol{B}^{(L)}. \tag{3.40}$$

Because we may define $\boldsymbol{A} = \boldsymbol{A}^{(1)}\boldsymbol{A}^{(2)}\cdots\boldsymbol{A}^{(L)}$, this hierarchical NMF is equal

to a standard NMF decomposition, $\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{B}^{(L)}$, and the simple hierarchical approach is thus not a different model but a specific procedure for computing a standard NMF. Cichocki and Zdunek report that each stage in the hierarchical procedure refines the solution, improves performance with respect to ill-conditioned or badly scaled data, and provides a mechanism for escaping local minima.

Another hierarchical NMF is presented by Ahn et al. [3] who introduce a transfer function between each layer of the hierarchical decomposition, such that $\boldsymbol{B}^{(l)} \approx g\big(\boldsymbol{A}^{(l+1)}\boldsymbol{B}^{(l+1)}\big)$, where g is an element-wise non-negative non-linear function. Ahn et al. discuss the similarities of this approach to a multilayer neural network. Hierarchical extensions to transformation invariant [16] and convolutive [188] NMF have also been proposed in the literature.

Relaxation of non-negativity

Ding et al. [56] propose a method denoted Semi-NMF in which the non-negativity constraint on $\boldsymbol{A}$ is relaxed

$$\boldsymbol{X} \approx \boldsymbol{C}\boldsymbol{B}, \quad \text{s.t. } \boldsymbol{B} \geq \boldsymbol{0}. \tag{3.41}$$

Here $\boldsymbol{C}$ (as well as the data matrix, $\boldsymbol{X}$, as usual) is allowed to take both positive and negative values.

Ding et al. [56] further propose an NMF model denoted Convex NMF where the columns of the $\boldsymbol{C}$ matrix are constrained to be convex combinations of the columns of the data matrix, $\boldsymbol{X}$

$$\boldsymbol{X} \approx \underbrace{\boldsymbol{X}\boldsymbol{A}}_{\boldsymbol{C}}\boldsymbol{B}, \tag{3.42}$$

where $\boldsymbol{A} \in \mathbb{R}_{+}^{J \times N}$. This model can be seen as a structured Semi-NMF method, where $\boldsymbol{C} = \boldsymbol{X}\boldsymbol{A}$ restricts the columns of $\boldsymbol{C}$ to lie inside the convex cone formed by the data. The method is closely related to the archetypal analysis algorithm of Cutler and Breiman [53].

Nonlinear and kernel NMF

Sra and Dhillon [209] discuss an NMF model where there is a non-linear relationship between the data $\boldsymbol{X}$ and the non-negative factorization, $\boldsymbol{A}\boldsymbol{B}$, modeled by a link function, h, such that

$$\boldsymbol{X} \approx h\left(\boldsymbol{A}\boldsymbol{B}\right), \tag{3.43}$$

where h is an element-wise function of its matrix argument.

Zhang et al. [242] introduce a kernel NMF method: Let $\phi(\boldsymbol{x})$ denote a non-linear function that maps a data vector into a high-dimensional (possibly infinite) feature space, and let $k(\boldsymbol{x}, \boldsymbol{y}) = \phi(\boldsymbol{x})^\top \phi(\boldsymbol{y})$ denote the inner product in the feature space which is also referred to as the kernel function. Then, the kernelized NMF method computes the non-negative decomposition in feature space

$$\phi(\boldsymbol{X}) \approx \boldsymbol{A}^{(\phi)} \boldsymbol{B}, \tag{3.44}$$

where $\boldsymbol{A}^{(\phi)}$ is a non-negative possibly infinite-dimensional feature matrix. Zhang et al. rewrite this expression as

$$\underbrace{\phi(\boldsymbol{X})^\top \phi(\boldsymbol{X})}_{\boldsymbol{K}} \approx \underbrace{\phi(\boldsymbol{X})^\top \boldsymbol{A}^{(\phi)}}_{\boldsymbol{Y}} \boldsymbol{B}, \tag{3.45}$$

where $\boldsymbol{K}_{j,j'} = k(\boldsymbol{X}_j, \boldsymbol{X}_{j'})$ and $\boldsymbol{Y}_{j,n} = \phi(\boldsymbol{X}_j)^\top \boldsymbol{A}_n^{(\phi)}$ where $\boldsymbol{X}_j$ denotes the jth column of $\boldsymbol{X}$. The authors proceed by solving $\boldsymbol{K} \approx \boldsymbol{Y}\boldsymbol{B}$ as a Semi-NMF problem.

Similarly, Ding et al. [56] kernelize the Convex NMF problem

$$\phi(\boldsymbol{X}) \approx \phi(\boldsymbol{X})\boldsymbol{A}\boldsymbol{B}, \tag{3.46}$$

and show that computation of the least squares cost function depends only on the kernel matrix $\boldsymbol{K}$.

3.3 Computing the NMF

NMF can be computed as a constrained optimization problem

$$\{\boldsymbol{A}, \boldsymbol{B}\} = \arg \min_{\boldsymbol{A},\boldsymbol{B} \geq \boldsymbol{0}} \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{B}), \tag{3.47}$$

based on the cost function $\mathcal{D}$. In principle, any constrained optimization algorithm can be used to compute $\boldsymbol{A}$ and $\boldsymbol{B}$. In the literature, many different algorithms have been proposed for NMF, many of which take advantage of the special structure of the NMF problem as well as properties of specific cost functions. Albright et al. [4], Berry et al. [22], and Sra and Dhillon [209] review a broad range of different algorithms.

Uniqueness

In general, NMF is not unique. Since cost functions are not jointly convex in the parameters $\boldsymbol{A}$ and $\boldsymbol{B}$, optimization algorithms can only at best guarantee convergence to a local minimum of the cost function. This means that several

runs of an algorithm on an NMF problem with different (random) initializations can result in different solutions. In practice, it can be useful to run an NMF algorithm several times and study the different solutions obtained. The uniqueness of NMF is further discussed by Donoho and Stodden [59] and Laurberg [123], and Theis et al. [214] discuss the uniqueness of sparse NMF.

Convergence

Iterative optimization algorithms compute a sequence of estimates, $\{\boldsymbol{A}, \boldsymbol{B}\}_{m=1}^{\infty}$, and for the algorithm to be convergent it must be guaranteed that the limit of the sequence is a local minimum of the cost function, i.e., it satisfies the Karush-Kuhn-Tucker (KKT) conditions.

If we denote the gradient of the cost function with respect to $\boldsymbol{A}$ and $\boldsymbol{B}$ by

$$\boldsymbol{\nabla}_{\boldsymbol{A}_{i,n}} = \frac{\partial \mathcal{D}(\boldsymbol{A}, \boldsymbol{B})}{\partial \boldsymbol{A}_{i,n}}, \quad \boldsymbol{\nabla}_{\boldsymbol{B}_{n,j}} = \frac{\partial \mathcal{D}(\boldsymbol{A}, \boldsymbol{B})}{\partial \boldsymbol{B}_{n,j}}, \tag{3.48}$$

the KKT necessary conditions for a solution to be locally optimal are

$$\begin{aligned} \boldsymbol{\nabla}_{\boldsymbol{A}}, \boldsymbol{\nabla}_{\boldsymbol{B}} &\geq \boldsymbol{0} && (3.49)\\ \boldsymbol{A}, \boldsymbol{B} &\geq \boldsymbol{0} && (3.50)\\ \boldsymbol{\nabla}_{\boldsymbol{A}} \otimes \boldsymbol{A}, \boldsymbol{\nabla}_{\boldsymbol{B}} \otimes \boldsymbol{B} &= \boldsymbol{0}, && (3.51) \end{aligned}$$

where $\otimes$ denotes the Hadamard (element-wise) matrix product. The KKT conditions state that at an optimal solution, the gradients as well as $\boldsymbol{A}$ and $\boldsymbol{B}$ are non-negative, and for each element in $\boldsymbol{A}$ and $\boldsymbol{B}$ either the gradient or the element's value is zero. The KKT conditions for the NMF problem can be written compactly as

$$\min\left(\boldsymbol{A}, \boldsymbol{\nabla}_{\boldsymbol{A}}\right) = \min\left(\boldsymbol{B}, \boldsymbol{\nabla}_{\boldsymbol{B}}\right) = \boldsymbol{0}, \tag{3.52}$$

where the minimum is taken element-wise. Gonzalez and Zhang [77] state that if a sequence converges to a local minimum, the residual norm of the KKT equation must go to zero, and propose to use the L_1 vector norm of the residual,

$$C_{\mathrm{KKT}} = \frac{1}{IN} \sum_{i,n} \left| \min(\boldsymbol{A}, \boldsymbol{\nabla}_{\boldsymbol{A}})_{i,n} \right| + \frac{1}{NJ} \sum_{n,j} \left| \min(\boldsymbol{B}, \boldsymbol{\nabla}_{\boldsymbol{B}})_{n,j} \right|, \tag{3.53}$$

to monitor the convergence. Another option is to use the maximum deviance from the KKT conditions which corresponds to the infinity norm. A detailed discussion of the optimality conditions for the least squares NMF is provided by Chu et al. [42].

3.3.1 Optimization strategies

Several optimization algorithms have been proposed in the literature, and these can be divided into three categories: direct optimization methods, alternating optimization methods, and alternating descent methods.

Direct optimization methods solve the NMF problem,

$$\{\boldsymbol{A}, \boldsymbol{B}\} \leftarrow \arg\min_{\boldsymbol{A},\boldsymbol{B}\geq\boldsymbol{0}} \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{B}), \tag{3.54}$$

directly using some (general-purpose) bound constrained optimization algorithm. In general, this is a non-negativity constrained non-linear optimization problem, for which many efficient algorithms exist. Since the number of parameters, $(I+J)N$, in the full standard NMF problem can be very high, it may be infeasible to use optimization methods that require the explicit computation of a Hessian matrix. An important and very useful method is the limited-memory Broyden-Fletcher-Goldfarb-Shanno method for bound constrained problems (L-BFGS-B) introduced by Byrd et al. [33, 34].

Alternating optimization methods partition the NMF problems into two subproblems for the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, that are solved in alternating turns until convergence,

$$\begin{aligned} &\textbf{repeat} \\ &\quad \boldsymbol{A} \leftarrow \arg\min_{\boldsymbol{A}\geq\boldsymbol{0}} \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{B}) \\ &\quad \boldsymbol{B} \leftarrow \arg\min_{\boldsymbol{B}\geq\boldsymbol{0}} \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{B}) \\ &\textbf{until } \text{convergence.} \end{aligned} \tag{3.55}$$

In each iteration, the NMF problem is solved for $\boldsymbol{A}$ while $\boldsymbol{B}$ is kept fixed and vice versa, and this is repeated until $\boldsymbol{A}$ and $\boldsymbol{B}$ converge to a solution of the full NMF problem. Bezdek et al. [25] analyze the convergence of alternating optimization[4] and show that under certain conditions the method will converge linearly to a local solution.

In general, alternating optimization may have several advantages [24] over direct optimization: when the parameters can be naturally partitioned into subsets for which efficient optimization algorithms exist, it can be faster than direct optimization; furthermore, alternating optimization can be better at avoiding local minima. Furthermore, if there are important differences between $\boldsymbol{A}$ and $\boldsymbol{B}$, such as one is sparse and the other is dense, or one is small and the other is large, it may be beneficial to solve the two subproblems with different tailored optimization algorithms.

[4] The authors refer to the method as *grouped variable coordinate descent*.

While the full NMF problem is not jointly convex in $\boldsymbol{A}$ and $\boldsymbol{B}$, some cost functions have the desirable property that the subproblems are convex in their respective parameters, which allows the computation of the globally optimal solution of each subproblem in each step. Also, for some cost functions, the rows of $\boldsymbol{A}$ (columns of $\boldsymbol{B}$) are decoupled when $\boldsymbol{B}$ ($\boldsymbol{A}$) is fixed which means that each subproblem consists of I (J) independent problems. As an example, for the least squares cost function the subproblems are sets of non-negativity constrained least squares problems that can be solved efficiently [29, 125].

Alternating descent methods relax the previously described approach by not computing an optimal solution for each subproblem in each step. Instead, an approximate solution is computed that reduces, but does not necessarily minimize, the cost function

$$\begin{aligned} &\textbf{repeat} \\ &\quad \boldsymbol{A} \leftarrow \boldsymbol{A}^* \quad \text{where } \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}^*, \boldsymbol{B}) < \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{B}) \\ &\quad \boldsymbol{B} \leftarrow \boldsymbol{B}^* \quad \text{where } \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{B}^*) < \mathcal{D}(\boldsymbol{X}; \boldsymbol{A}, \boldsymbol{B}) \\ &\textbf{until } \text{convergence.} \end{aligned} \tag{3.56}$$

This approach can be advantageous when an optimal solution of each subproblem can be computed by an iterative procedure where each iteration is fast and guaranteed to reduce the cost function. In this case, the method proceeds in turns by computing a single iteration on each subproblem. Although algorithms of this type reduce the cost function in each iteration, there is not in general any guarantee that the algorithm will converge to a local minimum of the NMF cost function. The multiplicative algorithms proposed by Lee and Seung [127] are examples of alternating descent NMF methods.

3.3.2 NMF algorithms

In this section a wide range of algorithms for the NMF problem are reviewed. A simplified vector notation,

$$\min_{\boldsymbol{x} \geq \boldsymbol{0}} f(\boldsymbol{x}), \tag{3.57}$$

is used to describe either the full NMF problem (3.54) or a sub-problem in an alternating optimization strategy (3.55–3.56).

Projected gradient descent

Lin [141] discusses the use of projected gradient descent methods for NMF, used for either direct or alternating optimization. Gradient descent is a simple

optimization strategy that searches for a local minimum of the cost function by iteratively taking steps in the direction of the negative gradient,

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \alpha \boldsymbol{\nabla}_f(\boldsymbol{x}) \,, \tag{3.58}$$

where α is a step size parameter and $\boldsymbol{\nabla}_f(\boldsymbol{x})$ is the gradient. Projected gradient methods extend the basic gradient descent by taking steps that are projected onto the feasible region (the non-negative orthant),

$$\boldsymbol{x} \leftarrow \max\left[\boldsymbol{x} - \alpha \boldsymbol{\nabla}_f(\boldsymbol{x}) \,, \boldsymbol{0}\right] . \tag{3.59}$$

The step size, α can, e.g., be chosen as a constant, by an adaptive procedure, or by line search. The step size can for example be chosen to yield the smallest value of the cost function that can be found in the gradient search direction,

$$\alpha = \arg\min_{\alpha^* \geq 0} f\big(\max\left[\boldsymbol{x} - \alpha^* \boldsymbol{\nabla}_f(\boldsymbol{x}) \,, \boldsymbol{0}\right]\big), \tag{3.60}$$

which is a minimization of function that is piecewise in α^*.

Liu et al. [145] presents a projected gradient descent algorithm for NMF that is based on the relative (natural) gradient. The algorithms for sparse NMF presented by Hoyer [96, 97] alternate between a projected gradient descent update for the sparse factor and a multiplicative update for the dense factor.

As an example, a simple projected gradient descent algorithm with fixed step size for the least squares NMF problem is given as Algorithm 1, where

$$R(x) = \begin{cases} x, & x > 0 \\ 0, & \text{otherwise,} \end{cases} \tag{3.61}$$

denotes projection onto the non-negative orthant.

Algorithm 1 Alternating least-squares projected gradient descent

Input: Step-size α, initial $\boldsymbol{A} \in \mathbb{R}_+^{I \times N}$ and $\boldsymbol{B} \in \mathbb{R}_+^{N \times J}$

1: **repeat**
2: $\quad \boldsymbol{A} \leftarrow R\big(\boldsymbol{A} - \alpha(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{X}\boldsymbol{B}^\top)\big)$
3: $\quad \boldsymbol{B} \leftarrow R\big(\boldsymbol{B} - \alpha(\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{B} - \boldsymbol{A}^\top\boldsymbol{X})\big)$
4: **until** convergence

Output: $\boldsymbol{A}$, $\boldsymbol{B}$

Multiplicative updates

Lee and Seung [127] present an iterative NMF algorithm with multiplicative updates, that can be seen as a rescaled gradient descent algorithm with a specific choice of step size. When the gradient can be expressed as the subtraction

of two non-negative terms, $\nabla_f(\boldsymbol{x}) = \nabla_f(\boldsymbol{x})^+ - \nabla_f(\boldsymbol{x})^-$, a step size can be chosen individually for each element of $\boldsymbol{x}$ as $\boldsymbol{\alpha}_i = \boldsymbol{x}_i / \nabla_f(\boldsymbol{x})_i^+$, which leads to a multiplicative gradient descent update

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i \frac{\nabla_f(\boldsymbol{x})_i^-}{\nabla_f(\boldsymbol{x})_i^+}. \tag{3.62}$$

Since this algorithm is formulated as a multiplication by a non-negative quantity, it is ensured that $\boldsymbol{x}$ remains non-negative, if it is initialized with positive elements. The initial value of $\boldsymbol{x}$ must be strictly positive, since any elements that are zero will remain zero in the following iterations. Lee and Seung [127] prove for the least squares and Kullback-Leibler divergences that the multiplicative updates are guaranteed to reduce the cost function in each step, and that the update rules are unity only at stationary points of the cost function. This does not imply, however, that the algorithm will converge to a stationary point within any reasonable number of iterations, as discussed by Gonzalez and Zhang [77] and Lin [142].

An accelerated Lee and Seung-type algorithm is proposed by Gonzalez and Zhang [77], who extend (3.62) by a step size scale parameter, β,

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i \beta \frac{\nabla_f(\boldsymbol{x})_i^-}{\nabla_f(\boldsymbol{x})_i^+}, \tag{3.63}$$

that is chosen to minimize the cost function in the rescaled gradient direction while ensuring that each step does not reduce any variable by more that a fixed fraction toward zero to avoid locking variables at the boundary of the feasible region. For the least squares cost function the authors derive a closed form expression for β, and for other cost functions the authors note that a line search may be required.

Multiplicative algorithms for sparse least squares NMF are proposed independently by Liu et al. [144] and Eggert and Körner [61]. The former algorithm includes an explicit normalization step, whereas the latter is based on a cost function that is invariant to normalization. Cichocki et al. [48, 49] present multiplicative algorithms for NMF with sparseness and smoothness constraints for a wide range of different cost functions.

As an example, the least squares multiplicative update algorithm is given as Algorithm 2.

Newton and quasi-Newton methods

Newton-type methods are based on approximating the cost function by a quadratic function, for which the optimum can be computed in closed form

Algorithm 2 Alternating least-squares multiplicative updates

Input: Initial $\boldsymbol{A} \in \mathbb{R}_+^{I \times N}$ and $\boldsymbol{B} \in \mathbb{R}_+^{N \times J}$

1: **repeat**

2: $\quad \boldsymbol{A}_{i,n} \leftarrow \boldsymbol{A}_{i,n} \dfrac{(\boldsymbol{X}\boldsymbol{B}^\top)_{i,n}}{(\boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^\top)_{i,n}}$

3: $\quad \boldsymbol{B}_{n,j} \leftarrow \boldsymbol{B}_{n,j} \dfrac{(\boldsymbol{A}^\top\boldsymbol{X})_{n,j}}{(\boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{B})_{n,j}}$

4: **until** convergence

Output: $\boldsymbol{A}$, $\boldsymbol{B}$

using the gradient and the Hessian, and leads to updates of the form

$$\boldsymbol{x} \leftarrow \boldsymbol{x} - \boldsymbol{H}_f^{-1}(\boldsymbol{x})\, \boldsymbol{\nabla}_f(\boldsymbol{x})\,, \tag{3.64}$$

where $\boldsymbol{H}_f(\boldsymbol{x})$ is the Hessian. In quasi-Newton methods, the Hessian is not computed explicitly but approximated, for example, using symmetric rank-1 updates or the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method.

When applying Newton-type methods to the NMF problem, special care must be taken to handle the non-negativity constraints, for example using a barrier function approach or an active set procedure: Simply projecting a Newton step onto the feasible region does not lead to a convergent algorithm, as discussed by Bertsekas [23] and Kim et al. [110].

Positive matrix factorization (PMF) is a weighted projected least squares algorithm proposed by Paatero and Tapper [159, 162]. The method is based on projected Newton updates, either in turns or simultaneously on $\boldsymbol{A}$ and $\boldsymbol{B}$, and the authors describe how to incorporate intermediate rotation steps to help eliminate negative elements in $\boldsymbol{A}$ or $\boldsymbol{B}$. As an alternative to the projection for handling the non-negativity constraints, the authors also discuss the use of a barrier function: A penalty term proportional to the squared value of negative elements. Lu and Wu [146] provide a detailed implementation guide for the PMF algorithm with a logarithmic barrier function.

Albright et al. [4] and Berry et al. [22] discuss the use of projected least squares, and argue that although the method is not theoretically well justified in terms of convergence, it is very useful in practice due to its speed and simplicity. Cichocki and Zdunek [45] present a weighted and regularized projected least squares algorithm for non-negative tensor factorization, and argue that the regularization and weighting terms can be utilized to improve the convergence properties of the algorithm.

As an example, a basic alternating projected least squares NMF algorithm is given as Algorithm 3.

Algorithm 3 Alternating projected least-squares Newton-update

Input: Initial $\boldsymbol{B} \in \mathbb{R}^{N \times J}$

1: **repeat**
2: $\quad \boldsymbol{A} \leftarrow R\big(\boldsymbol{X}\boldsymbol{B}^\top(\boldsymbol{B}\boldsymbol{B}^\top)^{-1}\big)$
3: $\quad \boldsymbol{B} \leftarrow R\big((\boldsymbol{A}^\top\boldsymbol{A})^{-1}\boldsymbol{A}^\top\boldsymbol{X}\big)$
4: **until** stop criterion

Output: $\boldsymbol{A}$, $\boldsymbol{B}$

Berry et al. [22] suggests that the least squares NMF problem can be solved in an alternating optimization approach using a least squares algorithm that properly handles the non-negativity constraints, such as the NNLS algorithm of Lawson and Hanson [125] or the fast NNLS proposed by Lawson and Hanson [125]. This approach leads to a convergent algorithm at the expense of a greatly increased computational cost [22].

Zdunek and Cichocki [239] present a projected quasi-Newton algorithm for NMF problems based on the Amari alpha family of divergence measures. The algorithm uses a Levenberg-Marquardt damped Newton update, and approximates the inverse Hessian using the Q-less QR factorization.

Another quasi-Newton approach is presented by Kim et al. [110] for the least squares NMF based on the BFGS approximation to the Hessian. The algorithm uses an active set procedure to handle the non-negativity constraints, and the authors demonstrate its use in an alternating optimization as well as an alternating descent strategy. A similar active set quasi-Newton method is proposed by Zdunek and Cichocki [238]. This algorithm alternates between a projected gradient step and a quasi-Newton step on the active set.

As an example of the active set approach, a simple least squares algorithm that in each iteration takes a Newton step on the active set is presented as Algorithm 4.

Other NMF algorithms

A simple method for enforcing non-negativity constraints is to re-parameterize the problem, $\boldsymbol{A} = f(\bar{\boldsymbol{A}})$, $\boldsymbol{B} = f(\bar{\boldsymbol{B}})$, using an element-wise function f that has the real numbers as its domain and the non-negative reals as its range. Thus, in the new set of parameters, $\bar{\boldsymbol{A}}$ and $\bar{\boldsymbol{B}}$, the NMF problem is an unconstrained optimization problem. Cichocki et al. [47] derive an algorithm based on multiplicative updates using the exponentiated gradient. They show that this algorithm corresponds to a gradient descent method in the space of the logarithm of the parameters $\boldsymbol{A}$ and $\boldsymbol{B}$.

Algorithm 4 Alternating least-squares active-set Newton-update

Input: Initial $\boldsymbol{A} \in \mathbb{R}_+^{I \times N}$ and $\boldsymbol{B} \in \mathbb{R}^{N \times J}$

repeat
  $\boldsymbol{\nabla_A} = \boldsymbol{A}\boldsymbol{B}\boldsymbol{B}^\top - \boldsymbol{X}\boldsymbol{B}^\top$
  for $i = 1$ to I **do**
    $\nu = \{n : \boldsymbol{A}_{i,n} \neq 0 \text{ or } \boldsymbol{\nabla_A}_{i,n} > 0\}$
    $\boldsymbol{A}_{i,\nu} \leftarrow R\left((\boldsymbol{X}\boldsymbol{B}^\top)_{i,\nu}\big((\boldsymbol{B}\boldsymbol{B}^\top)_{\nu,\nu}\big)^{-1}\right)$
  end for
  $\boldsymbol{\nabla_B} = \boldsymbol{A}^\top\boldsymbol{A}\boldsymbol{B} - \boldsymbol{A}^\top\boldsymbol{X}$
  for $j = 1$ to J **do**
    $\nu = \{n : \boldsymbol{B}_{n,j} \neq 0 \text{ or } \boldsymbol{\nabla_B}_{n,j} > 0\}$
    $\boldsymbol{B}_{\nu,j} \leftarrow R\left(\big((\boldsymbol{A}^\top\boldsymbol{A})_{\nu,\nu}\big)^{-1}(\boldsymbol{A}^\top\boldsymbol{X})_{i,\nu}\right)$
  end for
until convergence

Output: $\boldsymbol{A}$, $\boldsymbol{B}$

Heiler and Schnörr [88, 89] present an algorithm for the sparse least squares NMF problem and its tensor extension. The methods is based on alternating second order cone programming (SOCP), for which efficient large scale solvers exist, and the authors demonstrate that enforcing sparsity constraints fits nicely in this framework.

Chu and Lin [41] take a geometric approach to NMF: They view the problem as that of approximating the convex hull of a set of data points by a convex polytope on the probability simplex, and this leads to a geometrically inspired algorithm.

3.3.3 Initialization methods

Most algorithms for NMF are iterative and require initial values of $\boldsymbol{A}$ and $\boldsymbol{B}$, and many authors prescribe initializing $\boldsymbol{A}$ and $\boldsymbol{B}$ with random non-negative numbers. A suitably chosen initialization, however, can lead to faster convergence, and since the solution of most NMF problems is not unique, different initializations can lead to different solutions.

Because of the relation between NMF and clustering methods, and because of the notion that NMF finds a parts-based representation, it has been suggested to use simple clustering algorithms to compute a starting point for iterative NMF algorithms. Wild et al. [232] proposes to use the centroids from a spherical k-means clustering as initial values, and the method is reported to increase

the rate of convergence in the subsequent NMF. In the same spirit, Kim and Choi [117] present a greedy hierarchical clustering method based on a simple similarity measure.

Albright et al. [4] propose and compare several initialization strategies. A simple yet efficient method consists of computing the average of a random selection of data vectors. Another approach is based on projecting the N leading singular vectors onto the non-negative orthant, and the authors describe how fast initialization algorithms can be obtained using two well known approximations to the SVD.

Boutsidis and Gallopoulos [28] extends the idea of using the SVD as an initialization. They compute the first N singular vectors, which corresponds to approximating the data matrix by a sum of N rank-1 matrices. The authors proceed by approximating the non-negative elements of these N rank-1 matrices as N non-negative rank-1 matrices, and use these to initialize the non-negative matrix factorization. The method is reported to outperform random as well as spherical k-means initialization on several datasets.

Paper A

Non-negative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation

Mikkel N. Schmidt and Morten Mørup, "Non-negative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA), Lecture Notes in Computer Science (LNCS)*, Springer, vol. 3889, pp. 700–707, Apr. 2006.

Citations

This paper has been cited by FitzGerald et al. [66, 67], Gillet and Richard [76], Jang and Lee [102], Ozerov [156], Pearlmutter and Olsson [169], Raczyński et al. [174], Rennie [191], Smaragdis [206], Virtanen [224], and Zdunek and Cichocki [240, 241].

Nonnegative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation

Mikkel N. Schmidt and Morten Mørup

Technical University of Denmark
Informatics and Mathematical Modelling
Richard Petersens Plads, Building 321
DK-2800 Kgs. Lyngby, Denmark
{mns,mm}@imm.dtu.dk

Abstract. We present a novel method for blind separation of instruments in polyphonic music based on a non-negative matrix factor 2-D deconvolution algorithm. Using a model which is convolutive in both time and frequency we factorize a spectrogram representation of music into components corresponding to individual instruments. Based on this factorization we separate the instruments using spectrogram masking. The proposed algorithm has applications in computational auditory scene analysis, music information retrieval, and automatic music transcription.

1 Introduction

The separation of multiple sound sources from a single channel recording is a difficult problem which has been extensively addressed in the literature. Many of the proposed methods are based on matrix decompositions of a spectrogram representation of the sound. The basic idea is to represent the sources by different frequency signatures which vary in intensity over time.

Non-negative matrix factorization (NMF) [1, 2] has been proven a very useful tool in a variety of signal processing fields. NMF gives a sparse (or parts-based) decomposition [2] and under certain conditions the decomposition is unique [3] making it unnecessary to impose constraints in the form of orthogonality or independence. Efficient algorithms for computing the NMF have been introduced by Lee and Seung [4]. NMF has a variety of applications in music signal processing; recently, Helén and Virtanen [5] described a method for separating drums from polyphonic music using NMF and Smaragdis and Brown [6] used NMF for automatic transcription of polyphonic music.

When polyphonic music is modelled by factorizing the magnitude spectrogram with NMF, each instrument is modelled by an instantaneous frequency signature which can vary over time. Smaragdis [7] introduced an extension to NMF, namely the non-negative matrix factor deconvolution (NMFD) algorithm in which each instrument is modelled by a time-frequency signature which varies in intensity over time. Thus, the model can represent components with temporal structure. Smaragdis showed how this can be used to separate individual drums

from a real recording of drum sounds. This approach was further pursued by Wang and Plumley [8] who separated mixtures of different musical instruments. Virtanen [9] presented an algorithm based on similar ideas and evaluated its performance by separating mixtures of harmonic sounds.

In this paper, we propose a new method to factorize a log-frequency spectrogram using a model which can represent both temporal structure and the pitch change which occurs when an instrument plays different notes. We use a log-frequency spectrogram such that a pitch change corresponds to a displacement on the frequency axis. We denote this the non-negative matrix factor 2-D deconvolution (NMF2D). Where previous methods needed one component to model each note for each instrument, the proposed model represents each instrument compactly by a single time-frequency profile convolved in both time and frequency by a time-pitch weight matrix. This model dramatically decreases the number of components needed to model various instruments and effectively solves the blind single channel source separation problem for certain classes of musical signals. In section 2 we introduce the NMF2D model and derive the update equations for recursively computing the factorization based on two different cost functions. In section 3 we show how the algorithm can be used to analyze and separate polyphonic music and we compare the algorithm to the NMFD method of Smaragdis [7]. This is followed by a discussion of the results.

2 Method

Consider the non-negative matrix factorization problem:

$$\mathbf{V} \approx \mathbf{W}\mathbf{H}, \tag{1}$$

where $\mathbf{V}$,$\mathbf{W}$, and $\mathbf{H}$ are non-negative matrices. Lee and Seung [4] devise two algorithms to find $\mathbf{W}$ and $\mathbf{H}$: For the least square error and the KL divergence they show that the following recursive updates converge to a local minimum:

$$\begin{aligned} \text{Least square error}: \quad & \mathbf{W} \leftarrow \mathbf{W} \bullet \frac{\mathbf{V}\mathbf{H}^T}{\mathbf{W}\mathbf{H}\mathbf{H}^T}, \quad \mathbf{H} \leftarrow \mathbf{H} \bullet \frac{\mathbf{W}^T\mathbf{V}}{\mathbf{W}^T\mathbf{W}\mathbf{H}}, \\ \text{KL divergence}: \quad & \mathbf{W} \leftarrow \mathbf{W} \bullet \frac{\frac{\mathbf{V}}{\mathbf{W}\mathbf{H}}\mathbf{H}^T}{\mathbf{1}\cdot\mathbf{H}}, \quad \mathbf{H} \leftarrow \mathbf{H} \bullet \frac{\mathbf{W}^T\frac{\mathbf{V}}{\mathbf{W}\mathbf{H}}}{\mathbf{W}\cdot\mathbf{1}}, \end{aligned} \tag{2}$$

where $A \bullet B$ denotes element-wise multiplication and $\frac{A}{B}$ denotes element-wise division. These algorithms can be derived by minimizing the cost function using gradient descent and choosing the stepsize appropriately to yield simple multiplicative updates.

We now extend the NMF model to be a 2-dimensional convolution of $\mathbf{W}^\tau$ which depends on time, τ, and $\mathbf{H}^\phi$ which depends on pitch, ϕ. This forms the non-negative factor 2-D deconvolution (NMF2D) model:

$$\mathbf{V} \approx \mathbf{\Lambda} = \sum_\tau \sum_\phi \overset{\downarrow\phi}{\mathbf{W}^\tau} \overset{\rightarrow\tau}{\mathbf{H}^\phi}, \tag{3}$$

where $\downarrow \phi$ denotes the downward shift operator which moves each element in the matrix ϕ rows down, and $\rightarrow \tau$ denotes the right shift operator which moves each element in the matrix τ columns to the right, i.e.:

$$\mathbf{A} = \begin{pmatrix} 1\,2\,3 \\ 4\,5\,6 \\ 7\,8\,9 \end{pmatrix}, \quad \overset{\downarrow 2}{\mathbf{A}} = \begin{pmatrix} 0\,0\,0 \\ 0\,0\,0 \\ 1\,2\,3 \end{pmatrix}, \quad \overset{\rightarrow 1}{\mathbf{A}} = \begin{pmatrix} 0\,1\,2 \\ 0\,4\,5 \\ 0\,7\,8 \end{pmatrix}.$$

We note that the NMFD model introduced by Smaragdis [7] is a special case of the NMF2D model where $\phi = \{0\}$.

Each element in $\mathbf{\Lambda}$ is given by

$$\mathbf{\Lambda}_{i,j} = \sum_{\tau} \sum_{\phi} \sum_{d} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{H}^{\phi}_{d,j-\tau}. \tag{4}$$

In the following derivation of the update steps required to compute $\mathbf{W}^{\tau}$ and $\mathbf{H}^{\phi}$ we will need the derivative of a given element $\mathbf{\Lambda}_{i,j}$ with respect to a given element $\mathbf{W}^{\tau}_{k,d}$:

$$\frac{\partial \mathbf{\Lambda}_{i,j}}{\partial \mathbf{W}^{\tau}_{k,d}} = \frac{\partial}{\partial \mathbf{W}^{\tau}_{k,d}} \left(\sum_{\tau} \sum_{\phi} \sum_{d} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{H}^{\phi}_{d,j-\tau} \right) \tag{5}$$

$$= \frac{\partial}{\partial \mathbf{W}^{\tau}_{k,d}} \left(\sum_{\phi} \mathbf{W}^{\tau}_{i-\phi,d} \mathbf{H}^{\phi}_{d,j-\tau} \right) \tag{6}$$

$$= \begin{cases} \mathbf{H}^{\phi}_{d,j-\tau} & \phi = i-k \\ 0 & otherwise. \end{cases} \tag{7}$$

2.1 NMF2D Least Squares

Now, we consider the least squares cost function which corresponds to maximizing the likelihood of a gaussian noise model:

$$C_{LS} = ||\mathbf{V} - \mathbf{\Lambda}||_f^2 = \sum_{i} \sum_{j} (\mathbf{V}_{i,j} - \mathbf{\Lambda}_{i,j})^2. \tag{8}$$

Differentiating C_{LS} with respect to a given element in $\mathbf{W}^{\tau}$ gives:

$$\frac{\partial C_{LS}}{\partial \mathbf{W}^{\tau}_{k,d}} = \frac{\partial}{\partial \mathbf{W}^{\tau}_{k,d}} \sum_{i} \sum_{j} (\mathbf{V}_{i,j} - \mathbf{\Lambda}_{i,j})^2 \tag{9}$$

$$= -2 \sum_{i} \sum_{j} (\mathbf{V}_{i,j} - \mathbf{\Lambda}_{i,j}) \frac{\partial \mathbf{\Lambda}_{i,j}}{\partial \mathbf{W}^{\tau}_{k,d}} \tag{10}$$

$$= -2 \sum_{\phi} \sum_{j} (\mathbf{V}_{\phi+k,j} - \mathbf{\Lambda}_{\phi+k,j}) \mathbf{H}^{\phi}_{d,j-\tau}. \tag{11}$$

The recursive update steps for the gradient descent are given by:

$$\mathbf{W}_{k,d}^{\tau} \leftarrow \mathbf{W}_{k,d}^{\tau} - \eta \frac{\partial C_{LS}}{\partial \mathbf{W}_{k,d}^{\tau}}. \tag{12}$$

Similar to the approach of Lee and Seung [4], we choose the step size η so that the first term in (12) is canceled:

$$\eta = \frac{\mathbf{W}_{k,d}^{\tau}}{-2 \sum_{\phi} \sum_{j} \mathbf{\Lambda}_{\phi+k,j} \mathbf{H}_{d,j-\tau}^{\phi}}, \tag{13}$$

which gives us the following simple multiplicative updates:

$$\mathbf{W}_{k,d}^{\tau} \leftarrow \mathbf{W}_{k,d}^{\tau} \frac{\sum_{\phi} \sum_{j} \mathbf{V}_{\phi+k,j} \mathbf{H}_{d,j-\tau}^{\phi}}{\sum_{\phi} \sum_{j} \mathbf{\Lambda}_{\phi+k,j} \mathbf{H}_{d,j-\tau}^{\phi}}. \tag{14}$$

By noticing that transposing equation (3) interchanges the order of $\mathbf{W}^{\tau}$ and $\mathbf{H}^{\phi}$ in the model, the updates of $\mathbf{H}^{\phi}$ can easily be found. In matrix notation the updates can be written as:

$$\mathbf{W}^{\tau} \leftarrow \mathbf{W}^{\tau} \bullet \frac{\sum_{\phi} \overset{\uparrow \phi \rightarrow \tau}{\mathbf{V}}^{T} \mathbf{H}^{\phi}}{\sum_{\phi} \overset{\uparrow \phi \rightarrow \tau}{\mathbf{\Lambda}}^{T} \mathbf{H}^{\phi}} \qquad \mathbf{H}^{\phi} \leftarrow \mathbf{H}^{\phi} \bullet \frac{\sum_{\tau} \overset{\downarrow \phi}{\mathbf{W}^{\tau}}^{T} \overset{\leftarrow \tau}{\mathbf{V}}}{\sum_{\tau} \overset{\downarrow \phi}{\mathbf{W}^{\tau}}^{T} \overset{\leftarrow \tau}{\mathbf{\Lambda}}}. \tag{15}$$

2.2 NMF2D by KL Divergence

Consider the Kullbach-Leibler (KL) divergence given by:

$$C_{KL} = \sum_{i} \sum_{j} \mathbf{V}_{i,j} \log \frac{\mathbf{V}_{i,j}}{\mathbf{\Lambda}_{i,j}} - \mathbf{V}_{i,j} + \mathbf{\Lambda}_{i,j}. \tag{16}$$

Minimizing the KL divergence corresponds to assuming multinomial noise model. Differentiating this cost function with respect to a given element in $\mathbf{W}^{\tau}$ gives:

$$\frac{\partial C_{KL}}{\partial \mathbf{W}_{k,d}^{\tau}} = \frac{\partial}{\partial \mathbf{W}_{k,d}^{\tau}} \sum_{i} \sum_{j} \mathbf{V}_{i,j} \log \frac{\mathbf{V}_{i,j}}{\mathbf{\Lambda}_{i,j}} - \mathbf{V}_{i,j} + \mathbf{\Lambda}_{i,j} \tag{17}$$

$$= \sum_{i} \sum_{j} \left(1 - \frac{\mathbf{V}_{i,j}}{\mathbf{\Lambda}_{i,j}}\right) \frac{\partial \mathbf{\Lambda}_{i,j}}{\partial \mathbf{W}_{k,d}^{\tau}} \tag{18}$$

$$= \sum_{\phi} \sum_{j} \left(1 - \frac{\mathbf{V}_{k+\phi,j}}{\mathbf{\Lambda}_{k+\phi,j}}\right) \mathbf{H}_{d,j-\tau}^{\phi}. \tag{19}$$

Again, the recursive gradient descent update steps are given by:

$$\mathbf{W}_{k,d}^{\tau} \leftarrow \mathbf{W}_{k,d}^{\tau} - \eta \frac{\partial C_{KL}}{\partial \mathbf{W}_{k,d}^{\tau}}, \tag{20}$$

and the step size η is chosen so that the first term in equation (20) is canceled:

$$\eta = \frac{\mathbf{W}^{\tau}_{k,d}}{\sum_{\phi}\sum_{j}\mathbf{H}^{\phi}_{d,j-\tau}}, \tag{21}$$

which gives the following simple multiplicative updates:

$$\mathbf{W}^{\tau}_{k,d} \leftarrow \mathbf{W}^{\tau}_{k,d}\frac{\sum_{\phi}\sum_{j}\frac{\mathbf{V}_{\phi+k,j}}{\mathbf{\Lambda}_{\phi+k,j}}\mathbf{H}^{\phi}_{d,j-\tau}}{\sum_{\phi}\sum_{j}\mathbf{H}^{\phi}_{d,j-\tau}}. \tag{22}$$

Again, the updates for $\mathbf{H}^{\phi}$ can easily be found by symmetry, and the updates can be written in matrix notation as:

$$\mathbf{W}^{\tau} \leftarrow \mathbf{W}^{\tau} \bullet \frac{\sum_{\phi}\left(\frac{\mathbf{V}}{\mathbf{\Lambda}}\right)^{\overset{\uparrow\phi}{}}\overset{\rightarrow\tau}{\mathbf{H}^{\phi}}^{T}}{\sum_{\phi}\mathbf{1}\cdot\overset{\rightarrow\tau}{\mathbf{H}^{\phi}}^{T}} \qquad \mathbf{H}^{\phi} \leftarrow \mathbf{H}^{\phi} \bullet \frac{\sum_{\tau}\overset{\downarrow\phi}{\mathbf{W}^{\tau}}^{T}\left(\frac{\mathbf{V}}{\mathbf{\Lambda}}\right)^{\overset{\leftarrow\tau}{}}}{\sum_{\tau}\overset{\downarrow\phi}{\mathbf{W}^{\tau}}^{T}\cdot\mathbf{1}}. \tag{23}$$

3 Experimental Results

In order to demonstrate our NMF2D algorithm, we have analyzed a 4 second piece of computer generated polyphonic music containing a trumpet and a piano. For comparison we have also analyzed the same piece of music by the NMFD algorithm [7]. For both algorithms we used the least squares cost function. The score of the piece of music is shown in Fig. 1. The trumpet and the piano play a different short melodic passage each consisting of three distinct notes. We generated the music at a sample rate of 16 kHz and analyzed it by the short time Fourier transform with a 2048 point Hanning windowed FFT and 50% overlap. This gave us 63 FFT slices. We grouped the spectrogram bins into 175 logarithmically spaced frequency bins in the range of 50 Hz to 8 kHz with 24 bins per octave, which correponds to twice the resolution of the equal tempered musical scale. Then, we performed the NMF2D and NMFD factorization of the log-frequency magnitude spectrogram.

For the NMF2D we used two factors, $d = 2$, since we seek to separate two instruments. We empirically chose to use seven convolutive components in time, $\tau = \{0, \ldots, 6\}$, corresponding to approximatly 45 ms. The pitch of the notes played in the music span three whole notes. Consequently, we chose to use 12 convolutive components in pitch, i.e. $\phi = \{0, \ldots, 11\}$.

For the NMFD we used six factors, $d = 6$, corresponding to the total number of different tones played by the two instruments. Similar to the experiment with NMF2D we used seven convolutive components in time. For the experiment with NMFD we used our formulation of the NMF2D algorithm with $\phi = \{0\}$, since the NMFD is a special case of the NMF2D algorithm.

The results of the experiments with NMFD and NMF2D are shown in Fig. 2 and Fig. 3 respectively. The NMFD algorithm factorized each individual note from each instrument into a separate component, whereas the NMF2D algorithm factorized each instrument into a separate component.

We used the NMF2D factorization of the music to reconstruct the individual instruments separately by spectrogram masking. First, we reconstructed the spectrum of each individual instrument by computing equation (4) for each specific value of d. Based on these reconstructed individual instrument spectra we constructed a spectrogram mask for each instrument, so that each spectrogram bin is assigned to the instrument with the highest power at that bin. We mapped these spectrogram masks back into the linear-frequency spectrogram domain, filtered the complex spectrogram based on the masks, and computed the inverse filtered spectrogram using the original phase. The separation of the two instruments in the music is shown in Fig. 4. Informal listening test indicated, that the NMF2D algorithm was able to separate the two instruments very well.

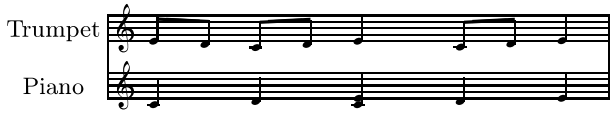

Fig. 1. Score of the piece of music used in the experiments. The music consists of a trumpet and a piano which play different short melodic passages each consisting of three distinct notes.

4 Discussion

In the previous section we compared the proposed NMF2D algorithm with NMFD. Both the NMF2D and the NMFD representation can be used to separate the instruments. However, since the notes of the individual instruments are spread over a number of factors in the NMFD, these must first be grouped manually or by other means. The NMF2D algorithm implicitly solves the problem of grouping notes.

If the assumption holds, that all notes for an instrument is an identical pitch shifted time-frequency signature, the NMF2D model will give better estimates of these signatures, because more examples (different notes) are used to compute each time-frequency signature. Even when this assumption does not hold, it migth still hold in a region of notes for an instrument. Furthermore, the NMF2D algorithm might be able to explain the spectral differences between two notes of different pitch by the 2-D convolution of the time-frequency signature.

Both the NMFD and NMF2D models perfectly explained the variation in the spectrogram. However, the number of free parameters in the two models are quite

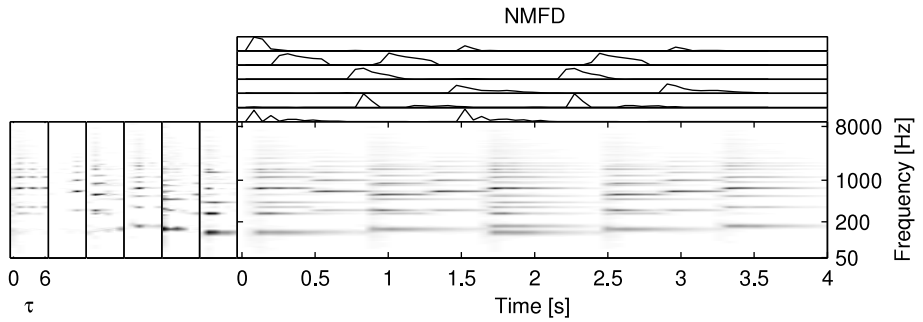

Fig. 2. Factorization of the piece of music using NMFD. The six time-frequency plots on the left are $\mathbf{W}^{\tau}$ for each factor, i.e. the time-frequency signature of the distincts tone played by the two instruments. The six plots on the top are the rows of $\mathbf{H}$ showing how the individual instrument notes are placed in time. The factors have been manually sorted so that the first three corresponds to the trumpet and the last three correspond to the piano.

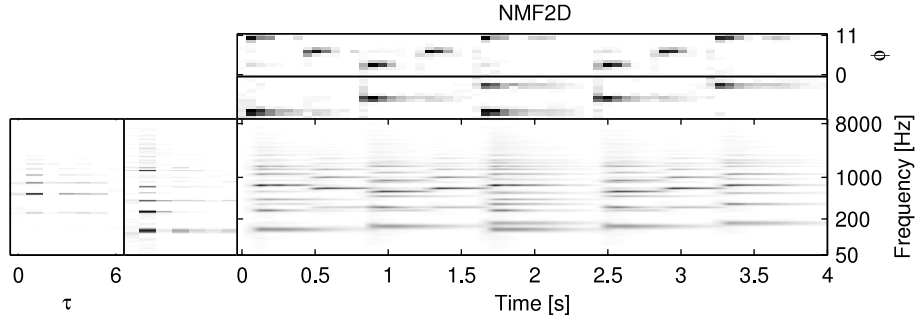

Fig. 3. Factorization of the piece of music using NMF2D. The two time-frequency plots on the left are $\mathbf{W}^{\tau}$ for each factor, i.e. the time-frequency signature of the two instruments. The two time-pitch plots on the top are $\mathbf{H}^{\phi}$ for each factor showing how the two instrument notes are placed in time and pitch.

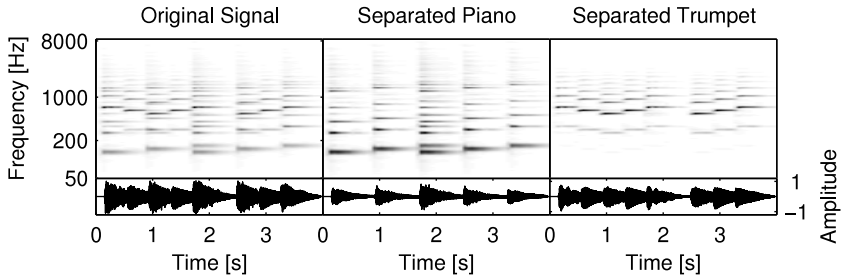

Fig. 4. Single channel source separation using NMF2D. The plots show the log-frequency spectrogram and the waveform of the music and the separated instruments.

different. If the dimensionality of the spectrogram is $I \times J$, and n_τ, n_ϕ denote the number of convolutive lags in time and pitch, NMFD has $(n_\tau I + J) \cdot d = (7 \cdot 175 + 63) \cdot 6 = 7728$ parameters whereas NMF2D has $(n_\tau I + n_\phi J) \cdot d = (7 \cdot 175 + 12 \cdot 63) \cdot 2 = 3962$ parameters. Consequently, the NMF2D was more restricted making the NMF2D the best model from an Occam's razor point of view.

Admittedly, the simple computer generated piece of music analyzed in this paper favors the NMF2D algorithm since each instrument key is a simple spectral shift of the same time-frequency signature. However, even when we analyze real music signals the NMF2D also gives very good results. Demonstrations of the algorithm for different music signals can be found at *www.intelligentsound.org*.

It is worth noting, that while we had problems making the NMFD algorithm converge in some situations when using the updates given by Smaragdis [7], the updates deviced in this paper to our knowledge always converge.

In the experiments above we used the NMF2D based on least squares. However, using the algorithm based on minimizing the KL divergence gave similar results. It is also worth mentioning that the NMF2D analysis is computationally inexpensive; the results in the previous section took approximatly 20 seconds to compute on a 2 GHz Pentium 4 computer.

It is our belief that the NMF2D algorithm can be useful in a wide range of areas including computational auditory scene analysis, music information retrieval, audio coding, automatic music transcription, and image analysis.

References

1. Paatero, P., Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(2) (1994) 111–126
2. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755) (1999) 788–91
3. Donoho, D., Stodden, V.: When does non-negative matrix factorization give a correct decomposition into parts? NIPS (2003)
4. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS. (2000) 556–562
5. Helén, M., Virtanen, T.: Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In: 13th European Signal Processing Conference. (2005)
6. Smaragdis, P., Brown, J.C.: Non-negative matrix factorization for polyphonic music transcription. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) (2003) 177–180
7. Smaragdis, P.: Non-negative matrix factor deconvolution; extraction of multiple sound sourses from monophonic inputs. International Symposium on Independent Component Analysis and Blind Source Separation (ICA) **3195** (2004) 494
8. Wang, B., Plumbley, M.D.: Musical audio stream separation by non-negative matrix factorization. In: Proceedings of the DMRN Summer Conference. (2005)
9. Virtanen, T.: Separation of sound sources by convolutive sparse cod. SAPA (2004)

Paper B

Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization

Mikkel N. Schmidt and Rasmus K. Olsson, "Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization," in *Spoken Language Processing, ICSLP International Conference on (INTERSPEECH)*, Sep. 2006.

Citations

This paper has been cited by Asari [7], Radfar and Dansereau [175, 176, 177], Pearlmutter and Olsson [169], Rennie et al. [192], and Virtanen [224].

Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization

Mikkel N. Schmidt and Rasmus K. Olsson
Informatics and Mathematical Modelling,
Technical University of Denmark
mns,rko@imm.dtu.dk

Abstract

We apply machine learning techniques to the problem of separating multiple speech sources from a single microphone recording. The method of choice is a sparse non-negative matrix factorization algorithm, which in an unsupervised manner can learn sparse representations of the data. This is applied to the learning of personalized dictionaries from a speech corpus, which in turn are used to separate the audio stream into its components. We show that computational savings can be achieved by segmenting the training data on a phoneme level. To split the data, a conventional speech recognizer is used. The performance of the unsupervised and supervised adaptation schemes result in significant improvements in terms of the target-to-masker ratio.

Index Terms: Single-channel source separation, sparse non-negative matrix factorization.

1 Introduction

A general problem in many applications is that of extracting the underlying sources from a mixture. A classical example is the so-called cocktail-party problem in which the problem is to recognize or isolate what is being said by an individual speaker in a mixture of speech from various speakers. A particular difficult version of the cocktail-party problem occurs when only a single-channel recording is available, yet the human auditory system solves this problem for us. Despite its obvious possible applications in, e.g., hearing aids or as a preprocessor to a speech recognition system, no machine has been built, which solves this problem in general.

Within the signal processing and machine learning communities, the single channel separation problem has been studied extensively, and different parametric and non-parametric signal models have been proposed.

Hidden Markov models (HMM) are quite powerful for modelling a single speaker. It has been suggested by Roweis [1] to use a factorial HMM to sepa-

rate mixed speech. Another suggestion by Roweis is to use a factorial-max vector quantizer [2]. Jang and Lee [3] use independent component analysis (ICA) to learn a dictionary for sparse encoding [4], which optimizes an independence measure across the encoding of the different sources. Pearlmutter and Olsson [5] generalize these results to overcomplete dictionaries, where the number of dictionary elements is allowed to exceed the dimensionality of the data. Other methods learn spectral dictionaries based on different types of non-negative matrix factorization (NMF) [6]. One idea is to assume a convolutive sum mixture, allowing the basis functions to capture time-frequency structures [7, 8].

A number researchers have taken ideas from the computational auditory scene analysis (CASA) literature, trying to incorporate various grouping cues of the human auditory system in speech separation algorithms [9, 10]. In the work by Ellis and Weiss [11] careful consideration is given to the representation of the audio signals so that the perceived quality of the separation is maximized.

In this work we propose to use the sparse non-negative matrix factorization (SNMF) [12] as a computationally attractive approach to sparse encoding separation. As a first step, overcomplete dictionaries are estimated for different speakers to give sparse representations of the signals. Separation of the source signals is achieved by merging the dictionaries pertaining to the sources in the mixture and then computing the sparse decomposition. We explore the significance of the degree of sparseness and the number of dictionary elements. We then compare the basic unsupervised SNMF with a supervised application of the same algorithm in which the training data is split into phoneme-level subproblems, leading to considerable computational savings.

2 Method

In the following, we consider modelling a magnitude spectrogram representation of a mixed speech signal. We represent the speech signal in the non-negative Mel spectrum magnitude domain, as suggested by Ellis and Weiss [11]. Here we posit that the spectrogram can be sparsely represented in an overcomplete basis,

$$\mathbf{Y} = \mathbf{DH} \tag{1}$$

that is, each data point held in the columns of $\mathbf{Y}$ is a linear combination of few columns of $\mathbf{D}$. The dictionary, $\mathbf{D}$, can hold arbitrarily many columns, and the code matrix, $\mathbf{H}$, is sparse. Furthermore, we assume that the mixture signal is a sum of R source signals

$$\mathbf{Y} = \sum_{i}^{R} \mathbf{Y}_i.$$

The basis of the mixture signal is then the concatenation of the source dictionaries, $\mathbf{D} = [\mathbf{D}_1 \dots \mathbf{D}_i \dots \mathbf{D}_R]$, and the complete code matrix is the concatenation of the source-individual codes, $\mathbf{H} = \left[\mathbf{H}_1^\top \dots \mathbf{H}_i^\top \dots \mathbf{H}_R^\top\right]^\top$. By enforcing the

sparsity of the code matrix, $\mathbf{H}$, it is possible to separate $\mathbf{Y}$ into its sources if the dictionaries are diverse enough.

As a consequence of the above, two connected tasks have to be solved: 1) the learning of source-specific dictionaries that yield sparse codes, and, 2) the computing of sparse decompositions for separation. We will use the sparse non-negative matrix factorization method proposed by Eggert and Körner [12] for both tasks.

2.1 Sparse Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) computes the decomposition in Equation (1) subject to the constraints that all matrices are non-negative, leading to solutions that are parts-based or sparse [6]. However, the basic NMF does not provide a well-defined solution in the case of overcomplete dictionaries, when the non-negativity constraints are not sufficient to obtain a sparse solution. The sparse non-negative matrix factorization (SNMF) optimizes the cost function

$$E \quad = \quad ||\mathbf{Y} - \bar{\mathbf{D}}\mathbf{H}||_F^2 + \lambda \sum_{ij} \mathbf{H}_{ij} \quad \text{s.t.} \quad \mathbf{D}, \mathbf{H} \geq \mathbf{0} \tag{2}$$

where $\bar{\mathbf{D}}$ is the column-wise normalized dictionary matrix. This cost function is the basic NMF quadratic cost augmented by an L_1 norm penalty term on the coefficients in the code matrix. The parameter, λ, controls the degree of sparsity. Any method that optimizes Equation (2) can be regarded as computing a maximum posterior (MAP) estimate given a Gaussian likelihood function and a one-sided exponential prior distribution over $\mathbf{H}$. The SNMF can be computed by alternating updates of $\mathbf{D}$ and $\mathbf{H}$ by the following rules [12]

$$\begin{aligned} \mathbf{H}_{ij} &\leftarrow \mathbf{H}_{ij} \bullet \frac{\mathbf{Y}_i^\top \bar{\mathbf{D}}_j}{\mathbf{R}_i^\top \bar{\mathbf{D}}_j + \lambda} \\ \mathbf{D}_j &\leftarrow \mathbf{D}_j \bullet \frac{\sum_i \mathbf{H}_{ij} \left[\mathbf{Y}_i + (\mathbf{R}_i^\top \bar{\mathbf{D}}_j)\bar{\mathbf{D}}_j\right]}{\sum_i \mathbf{H}_{ij} \left[\mathbf{R}_i + (\mathbf{V}_i^\top \bar{\mathbf{D}}_j)\bar{\mathbf{D}}_j\right]} \end{aligned}$$

where $\mathbf{R} = \mathbf{DH}$, and the bold operators indicate pointwise multiplication and division.

We first apply SNMF to learn dictionaries of individual speakers. To separate speech mixtures we keep the dictionary fixed and update only the code matrix, $\mathbf{H}$. The speech is then separated by computing the reconstruction of the parts of the sparse decomposition pertaining to each of the used dictionaries.

2.2 Two Ways to Learn Sparse Dictionaries

We study two approaches to learning sparse dictionaries, see Figure 1. The first is a direct, unsupervised approach where the dictionary is learned by computing the SNMF on a large training data set of a single speaker. The second approach

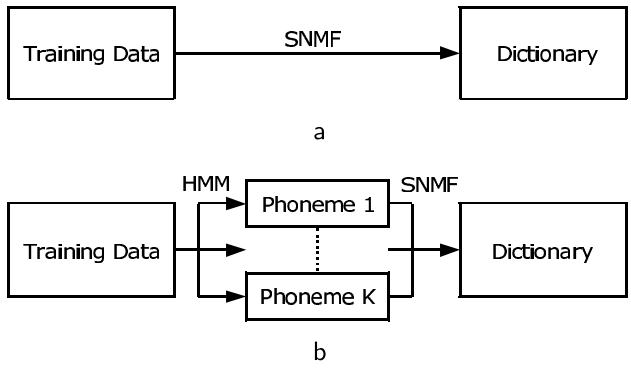

Figure 1: Two approaches for learning sparse dictionaries of speech. The first approach (a) is to learn the dictionary from a sparse non-negative matrix factorization of the complete training data. The second approach (b) is to segment the training data into individual phonemes, learn a sparse dictionary for each phoneme, and compute the dictionary by concatenating the individual phoneme dictionaries.

is to first segment the training data according to phoneme labels obtained by speech recognition software based on a hidden Markov model. Then, a sparse dictionary is learned for each phoneme and the final dictionary is constructed by concatenating the individual phoneme dictionaries. As a consequence, a smaller learning problem is addressed by the SNMF for each of the phonemes.

The computational savings associated with this divide-and-conquer approach are significant. Since the running time of the SNMF scales with the size of the training data and the number of elements in the dictionary, dividing the problem into SNMF subproblems for each phoneme reduces the overall computational burden by a factor corresponding to the number of phonemes. For example, if the data is split into 40 phonemes, we need to solve 40 SNMF subproblems each with a complexity of $1/40^2$ compared to the full SNMF problem. In addition to this, since the phoneme SNMF subproblems are much smaller than the total SNMF problem, a faster convergence of the iterative SNMF algorithm can be expected. These advantages makes it desirable to compare the quality of sparse dictionaries estimated by the two methods.

3 Simulations

Part of the Grid Corpus [13] was used for evaluating the proposed method for speech separation. The Grid Corpus consists of simple structured sentences from a small vocabulary, and has 34 speakers and 1000 sentences per speaker. Each utterance is a few seconds and word level transcriptions are available. We used half of the corpus as a training set.

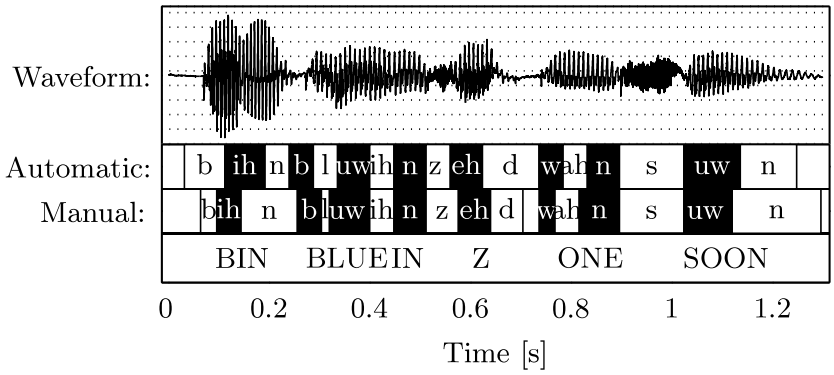

Figure 2: The automatic phoneme transcription as computed by the trained hidden Markov model (HMM) for an example sentence from the Grid Corpus. A manual transcription is provided for comparison, confirming the conventional hypothesis that the HMM is a useful tool in segmenting a speech signal into its phonemes.

3.1 Phoneme Transcription

First, we used speech recognition software to generate phoneme transcriptions of the sentences. For each speaker in the corpus a phoneme-based hidden Markov model (HMM) was trained using the HTK toolkit[1]. The HMM's were used to compute an alignment of the phonemes in each sentence, taking the pronunciations of each word from the British English Example Pronunciation (BEEP) dictionary[2]. This procedure provided phoneme-level transcriptions of each sentence. In order to evaluate the quality of the phoneme alignment, the automatic phoneme transcription was compared to a manual transcription for a few sentences. We found that the automatic phoneme alignment in general was quite reasonable. An example is given in Figure 2.

3.2 Preprocessing and Learning Dictionaries

We preprocessed the speech data in a similar fashion to Ellis and Weiss [11]: the speech was prefiltered with a high-pass filter, $1 - 0.95z^{-1}$, and the STFT was computed with an analysis window of 32ms at a sample rate of 25kHz. An overlap of 50 percent was used between frames. This yielded a spectrogram with 401 frequency bins which was then mapped into 80 frequency bins on the Mel scale. The training set was re-weighted so that all frames containing energy above a threshold were normalized by their standard deviation. The resulting magnitude Mel-scale spectrogram representation was employed in the experiments.

[1] Avaiable from htk.eng.cam.ac.uk.

[2] Available by anonymous ftp from svr-ftp.eng.cam.ac.uk/pub/comp.speech/dictionaries/beep.tar.gz.

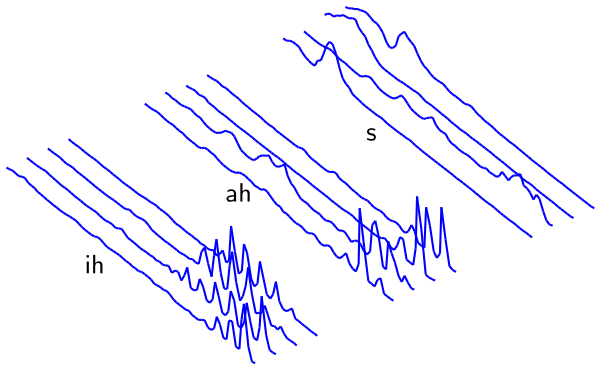

Figure 3: A few samples of columns of phoneme dictionaries learned from female speech. The SNMF was applied to data, which had been phoneme-labelled by a speech recognizer. Not surprisingly, the basis functions exhibit the some general properties of the respective phonemes, and additional variation is captured by the algorithm, such as the fundamental frequency in the case of voiced phonemes.

In order to assess the effects of the model hyper-parameters and the effect of splitting the training data according the phoneme transcriptions, a subset of four male and four female speakers were extracted from the Grid Corpus. We constructed a set of 64 mixed sentences by mixing two randomly selected sentences for all combinations of the eight selected test speakers.

Two different sets of dictionaries were estimated for each speaker. The first set was computed by concatenating the spectrograms for each speaker and computing the SNMF on the complete training data for that speaker. The second set was computed by concatenating the parts of the training data corresponding to each phoneme for each speaker, computing the SNMF for each phoneme spectrogram individually, and finally concatenating the individual phoneme dictionaries. To save computation, only 10 percent of the training set was used to train the dictionaries. In a Matlab environment running on a 1.6GHz Intel processor the computation of the SNMF for each speaker took approximately 30 minutes, whereas the SNMFs for individual phonemes were computed in a few seconds. The algorithm was allowed to run for maximally 500 iterations or until convergence as defined by the relative change in the cost function. Figure 3 shows samples from a dictionary which was learned using SNMF on the phoneme-segmented training data for a female speaker. The dictionaries were estimated for four different levels of sparsity, $\lambda = \{0.0001, 0.001, 0.01, 0.1\}$, and four different dictionary sizes, $N = \{70, 140, 280, 560\}$. This was done for both the complete and the phoneme-segmented training data.

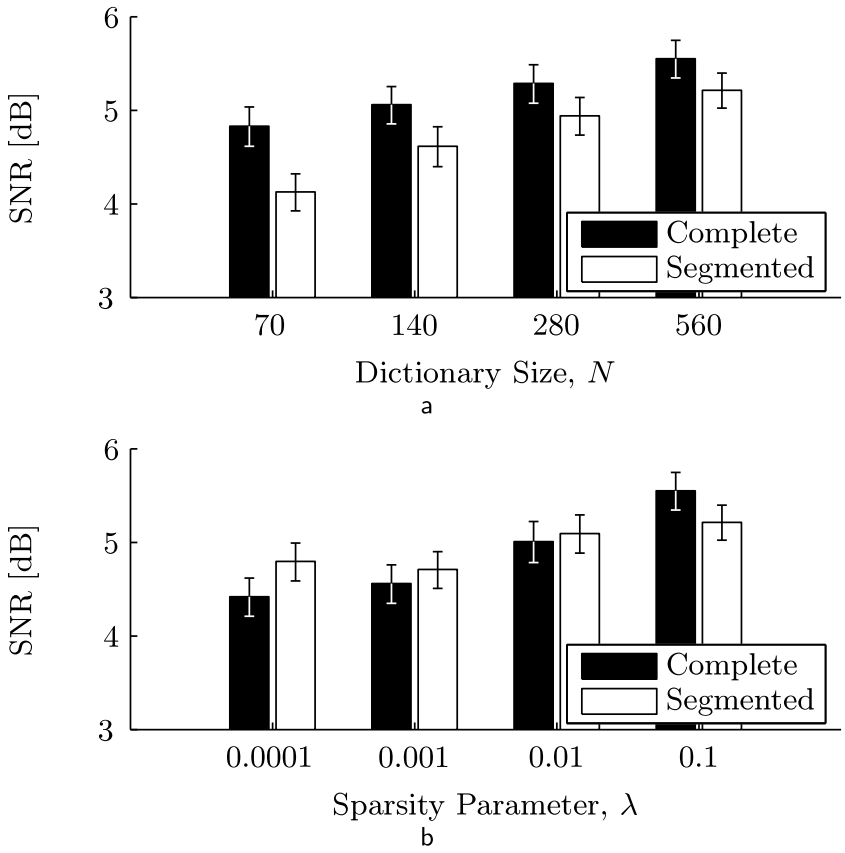

Figure 4: Average signal-to-noise ratio (SNR) of the separated signals for dictionaries trained on the complete speech spectrograms and on individual phonemes, (a) as a function of the dictionary size, N, with sparsity $\lambda = 0.1$, and (b) as a function of the sparsity with $N = 560$. We found that the SNMF algorithm did not give useful results when $\lambda = 1$.

	Complete	Segmented
Same gender	4.8±0.4 dB	4.3±0.3 dB
Opp. gender	6.6±0.3 dB	6.4±0.3 dB

Table 1: Average signal-to-noise ratio (SNR) of the separated signals for dictionaries trained on the complete speech spectrograms and on individual phonemes. Dictionaries were learned with $N = 560$ and $\lambda = 0.1$.

3.3 Speech Separation

For each test sentence, we concatenated the dictionaries of the two speakers in the mixture, and computed the code matrix using the SNMF updates. Then, we reconstructed the individual magnitude spectra of the two speakers and mapped them from the Mel-frequency domain into the linear frequency STFT domain. Separated waveforms were computed by spectral masking and spectrogram inversion, using the original phase of the mixed signal. The separated waveforms were then compared with the original clean signals, computing the signal-to-noise ratio.

The results in Figure 4 show that the quality of separation increases with N. This agrees well with the findings of Ellis and Weiss [11]. Furthermore, the choice of sparsity, λ, is important for the performance of the separation method, especially in the case of unsegmented data. The individual phoneme-level dictionaries are so small in terms of N that the gain from enforcing sparsity in the NMF is not as significant; the segmentation in itself sparsifies the dictionary to some extend. Table 1 shows that the method works best for separating speakers of opposite gender, as would be expected. Audio examples are available at mikkelschmidt.dk/interspeech2006 .

3.4 Interspeech 2006: Speech Separation Challenge

We evaluated the algorithm on the Speech Separation test set, which was constructed by adding a target and a masking speaker at different target-to-masker ratios (TMR)[3]. As an evaluation criterion, the word-recognition rate (WRR) for the letter and number in the target speech signal was computed using the HTK speech recognizer trained on data separated by the proposed method. A part of the test was to blindly identify the target signal as the one separated signal, which containing the word 'white'. A total of 600 mixtures were evaluated for each TMR. The source signals were separated and reconstructed in the time-domain as described previously. In Table 2, the performance of the method is contrasted with the performance of human listeners [14]. A subtask in obtaining these results was to estimate the identities of the speakers in the mixtures. This was done by exhaustively applying the SNMF to the signals with all pairs of two dictionaries, selecting the combination that gave the best

[3]This test set is due to Cooke and Lee. It is available at http://www.dcs.shef.ac.uk/ martin/SpeechSeparationChallenge.htm.

TMR	6dB	3dB	0dB	−3dB	−6dB	−9dB
Human Performance						
ST	90%	72%	54%	52%	60%	68%
SG	93%	85%	76%	72%	77%	80%
DG	94%	91%	86%	88%	87%	83%
All	92%	83%	72%	71%	75%	77%
Proposed Method						
ST	56%	53%	45%	38%	31%	28%
SG	60%	57%	52%	44%	37%	32%
DG	73%	72%	71%	63%	54%	41%
All	64%	62%	58%	51%	42%	35%

Table 2: Results from applying the SNMF to the Speech Separation Challenge: the word-recognition rate (WRR) on separated mixtures of speech in varying target-masker ratios (TMR) in same talker (ST), same gender (SG) different gender (DG), and overall (All) conditions compared with human performance on the mixtures. The WRR should be compared to that of other algorithms applied to the same test set (see the conference proceedings).

fit. We are currently investigating methods to more efficiently determine the active sources in a mixture.

4 Discussion and Outlook

We have successfully applied sparse non-negative matrix factorization (SNMF) to the problem of monaural speech separation. The SNMF learns large over-complete dictionaries, leading to a more sparse representations of individual speakers than for example the basic NMF. Inspection of the dictionaries reveals that they capture fundamental properties of speech, in fact they learn basis functions that resemble phonemes. This has lead us to adopt a working hypothesis that the learning of signal dictionaries on a phoneme level is a computational shortcut to the goal, leading to similar performance. Our experiments show that the practical performance of sparse dictionaries learned in this way performs only slightly worse than dictionaries learned on the complete dataset. In future work, we hope to benefit further from the phoneme labelling of the dictionaries in formulating transitional models in the encoding space of the SNMF, hopefully matching the dynamics of speech.

5 Acknowledgements

This work made possible in part by funding from Oticon Fonden. We would like to thank Lars Kai Hansen and Jan Larsen for fruitful discussions, and acknowledge Dan Ellis for making available useful software at his homepage.

References

[1] S. T. Roweis, "One microphone source separation," in *NIPS*, 2001, pp. 793–799.

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

[3] G. J. Jang and T. W. Lee, "A maximum likelihood approach to single channel source separation," *JMLR*, vol. 4, pp. 1365–1392, 2003.

[4] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[5] B. A. Pearlmutter and R. K. Olsson, "Algorithmic differentiation of linear programs for single-channel source separation," in *MLSP, submitted*, 2006.

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

[7] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," in *SAPA*, 2004.

[8] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," in *ICA*, 2005.

[9] B. A. Pearlmutter and A. M. Zador, "Monaural source separation using spectral cues," in *ICA*, 2004, pp. 478–485.

[10] F. Bach and M. I. Jordan, "Blind one-microphone speech separation: A spectral learning approach," in *NIPS*, 2005, pp. 65–72.

[11] D. P. W. Ellis and R. J. Weiss, "Model-based monaural source separation using a vector-quantized phase-vocoder representation," in *ICASSP*, 2006.

[12] J. Eggert and E. Körner, "Sparse coding and nmf," in *Neural Networks*. 2004, vol. 4, pp. 2529–2533, IEEE.

[13] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *submitted to JASA*.

[14] M. P. Cooke, M. L. Garcia Lecumberri, and J. Barker, "The non-native cocktail party (in preparation)," .

Paper C

Wind Noise Reduction using Non-negative Sparse Coding

Mikkel N. Schmidt, Jan Larsen, and Fu-Tien Hsiao, "Wind Noise Reduction using Non-negative Sparse Coding," in *Machine Learning for Signal Processing, IEEE International Workshop on (MLSP)*, pp. 431–436, Aug. 2007.

Citations

This paper has been cited by Laurberg et al. [124].

Wind Noise Reduction using Non-Negative Sparse Coding

Mikkel N. Schmidt, Jan Larsen
Technical University of Denmark
Informatics and Mathematical Modeling
Richard Petersens Plads, Building 321
2800 Kgs. Lyngby

Fu-Tien Hsiao
IT University of Copenhagen
Multimedia Technology
Rued Langgaards Vej 7
2300 Copenhagen S.

Abstract

We introduce a new speaker independent method for reducing wind noise in single-channel recordings of noisy speech. The method is based on non-negative sparse coding and relies on a wind noise dictionary which is estimated from an isolated noise recording. We estimate the parameters of the model and discuss their sensitivity. We then compare the algorithm with the classical spectral subtraction method and the Qualcomm-ICSI-OGI noise reduction method. We optimize the sound quality in terms of signal-to-noise ratio and provide results on a noisy speech recognition task.

1 Introduction

Wind noise can be a major problem in outdoor recording and processing of audio. A good solution can be to use a high quality microphone with a wind screen; this is not possible, however, in applications such as hearing aids and mobile telephones. Here, we typically have available only a single-channel recording made using an unscreened microphone. To overcome the wind noise problem in these situations, we can process the recorded signal to reduce the wind noise and enhance the signal of interest. In this paper, we deal with the problem of reducing wind noise in single-channel recordings of speech.

There exists a number of methods for noise reduction and source separation. When the signal of interest and the noise have different frequency characteristics, the Wiener filter is a good approach to noise reduction. The idea is to attenuate the frequency regions where the noise is dominant. In the case of speech and wind noise, however, this approach leads only to limited performance, since both speech and wind noise are non-stationary broad-band signals with most of the energy in the low frequency range as shown in Figure 1.

Another widely used approach is spectral subtraction [1]. Here, the idea is to subtract an estimate of the noise spectrum from the spectrum of the mixed signal. Spectral subtraction takes advantage of the non-stationarity of the speech

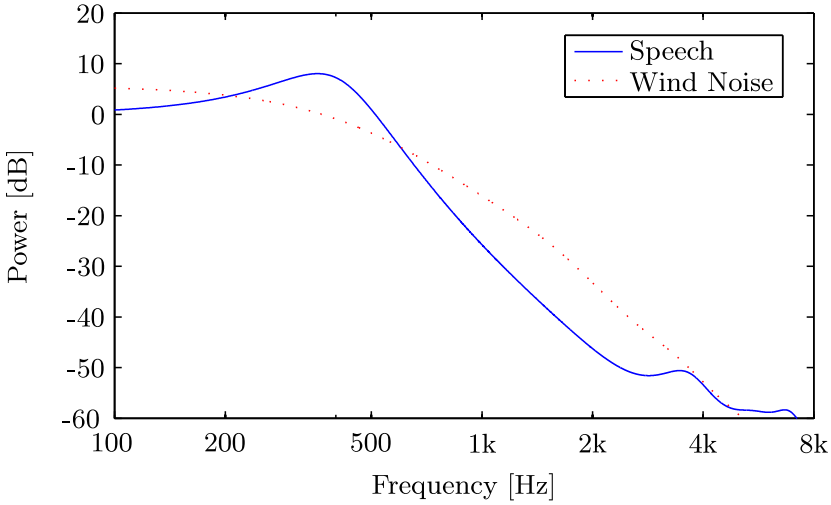

Figure 1: Average spectrum of speech and wind noise. Both speech and wind noise are broad-band signals with most of the energy in the low frequency range. The spectra are computed using the Burg method based on a few seconds of recorded wind noise and a few seconds of speech from eight different speakers.

signal by reestimating the noise spectrum when there is no speech activity. During speech activity, the noise is assumed stationary, and for this reason the method is best suited for situations where the noise varies slowly compared to the speech. This is not the case for wind noise. As illustrated in Figure 2, wind noise changes rapidly and wind gusts can have very high energy.

A number of methods for separating non-stationary broad-band signals based on source modeling have been proposed. The idea is to first model the sources independently and then model the mixture using the combined source models. Finally, the sources can be reconstructed individually for example by refiltering the mixed signal. Different models for the sources have been proposed, such as a hidden Markov model with a Gaussian mixture model [2], vector quantization [3, 4], and non-negative sparse coding [5]. A limitation of these approaches is that each source must be modeled prior to the separation. In the case of wind noise reduction, this means that we must model both the speech and the wind noise beforehand.

Binary spectral masking is a source separation method, where the main assumption is that the sources can be separated by multiplying the spectrogram by a binary mask. This is reasonable when each time-frequency bin is dominated by only one source. Thus, the problem of separating signals is reduced to that of estimating a binary time-frequency mask. One approach to estimating the mask is to use a suitable classification technique such as the relevance vector machine [6]. Similar to the source modeling approach, however, both the sources must be known in advance in order to estimate the parameters of the classifier.

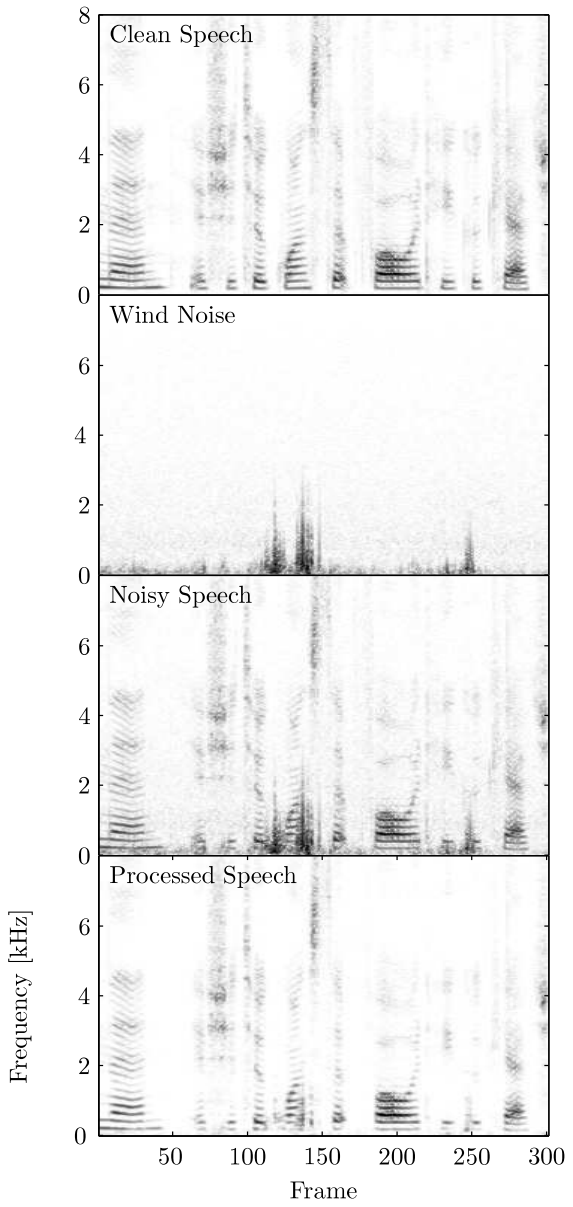

Figure 2: Example spectrograms and the result of the algorithm. Spectrograms of clean speech and wind noise: Both speech and wind noise are non-stationary broad-band signals. Speech has both harmonic and noise-like segments and sometimes short pauses between words. Wind noise is characterized by a constant broad-band background noise and high energy broad-band wind gusts. There is a large overlap between the speech and noise in the noisy recording. In the processed signal, a large part of the noise is removed.

A completely different approach to source separation is computational auditory scene analysis (CASA). Here, the idea is to simulate the scene analysis process performed by the human auditory system. We will not discuss this further in this paper.

2 Method

In this work, we propose a new method for noise reduction, which is related to the source modeling approach using non-negative sparse coding. The key idea is to build a speaker independent system, by having a source model for the wind noise but not for the speech.

We assume that the speech signal and the wind noise are additive in the time domain, i.e., we assume that the noise is not so strong, that we have problems with saturation. Then, the noisy signal, $x(t)$, can be written as

$$x(t) = s(t) + n(t), \tag{1}$$

where $s(t)$ is the speech signal, and $n(t)$ is the wind noise. If we assume that the speech and wind noise are uncorrelated, this linearity applies in the power spectral domain as well.

In line with Berouti et al. [7], we represent the signal in the time-frequency domain as an element wise exponentiated short time Fourier transform

$$\boldsymbol{X} = |\mathrm{STFT}\{x(t)\}|^{\gamma}. \tag{2}$$

When the exponent, γ, is set to 2 the representation is the power spectrogram and the above mentioned linearity holds on average. Although using $\gamma \neq 2$ violates the linearity property, it often leads to better performance; in the sequel, we estimate a suitable value for this parameter.

2.1 Non-negative sparse coding

The idea in non-negative sparse coding (NNSC) is to factorize the signal matrix as

$$\boldsymbol{X} \approx \boldsymbol{D}\boldsymbol{H}, \tag{3}$$

where $\boldsymbol{D}$ and $\boldsymbol{H}$ are non-negative matrices which we refer to as the dictionary and the code. The columns of the dictionary matrix constitute a source specific basis and the sparse code matrix contains weights that determine by which amplitude each element of the dictionary is used in each time frame. It has been shown that imposing non-negativity constraints leads to a parts-based representation, because only additive and not subtractive combinations are allowed [8]. Enforcing sparsity of the code leads to solutions where only a few dictionary elements are active simultaneously. This can lead to better solutions, because it forces the dictionary elements to be more source specific.

There exists different algorithms for computing this factorization [9, 10, 11, 12]. In the following we use the method proposed by Eggert and Körner [10],

which is perhaps not the most efficient method, but it has a very simple formulation and allows easy implementation. The NNSC algorithm starts with randomly initialized matrices, $\boldsymbol{D}$ and $\boldsymbol{H}$, and alternates the following updates until convergence

$$\boldsymbol{H} \leftarrow \boldsymbol{H} \bullet \frac{\bar{\boldsymbol{D}}^\top \boldsymbol{X}}{\bar{\boldsymbol{D}}^\top \bar{\boldsymbol{D}} \boldsymbol{H} + \lambda}, \tag{4}$$

$$\boldsymbol{D} \leftarrow \bar{\boldsymbol{D}} \bullet \frac{\boldsymbol{X}\boldsymbol{H}^\top + \bar{\boldsymbol{D}} \bullet \big(\boldsymbol{1}\big(\bar{\boldsymbol{D}}\boldsymbol{H}\boldsymbol{H}^\top \bullet \bar{\boldsymbol{D}}\big)\big)}{\bar{\boldsymbol{D}}\boldsymbol{H}\boldsymbol{H}^\top + \bar{\boldsymbol{D}} \bullet \big(\boldsymbol{1}\big(\boldsymbol{X}\boldsymbol{H}^\top \bullet \bar{\boldsymbol{D}}\big)\big)}. \tag{5}$$

Here, $\bar{\boldsymbol{D}}$ is the columnwise normalized dictionary matrix, $\boldsymbol{1}$ is a square matrix of suitable size with all elements equal to 1, and the bold operators indicate pointwise multiplication and division. The parameter λ determines the degree of sparsity in the code matrix.

2.2 Non-negative sparse coding of a noisy signal

When the sparse coding framework is applied to a noisy signal and we assume that the sources are additive, we have

$$\boldsymbol{X} = \boldsymbol{X}_s + \boldsymbol{X}_n \approx [\boldsymbol{D}_s \ \boldsymbol{D}_n] \begin{bmatrix} \boldsymbol{H}_s \\ \boldsymbol{H}_n \end{bmatrix} = \boldsymbol{D}\boldsymbol{H}, \tag{6}$$

where the subscripts, s and n, indicate speech and noise. Inherent in the sparse coding approach, however, is a permutation ambiguity; the order of the columns of $\boldsymbol{D}$ can be changed as long as the rows of $\boldsymbol{H}$ are changed correspondingly. Consequently, we need a mechanism to fix or determine which components pertain to which source. One method is to precompute the source dictionaries using isolated recordings of the sources [5]. Another idea is to devise an automatic grouping rule as argued by Wang and Plumbley [14]. We suggest to precompute the source dictionary for only one of the sources, the wind noise, and to learn the dictionary of the speech directly from the noisy data. This results in a method which is independent of the speaker.

We modify the NNSC algorithm so that only $\boldsymbol{D}_s$, $\boldsymbol{H}_s$, and $\boldsymbol{H}_n$ are updated. This gives us the following update equations

$$\boldsymbol{H}_s \leftarrow \boldsymbol{H}_s \bullet \frac{\bar{\boldsymbol{D}}_s^\top \boldsymbol{X}}{\bar{\boldsymbol{D}}_s^\top \bar{\boldsymbol{D}} \boldsymbol{H} + \ell_s}, \qquad \boldsymbol{H}_n \leftarrow \boldsymbol{H}_n \bullet \frac{\bar{\boldsymbol{D}}_n^\top \boldsymbol{X}}{\bar{\boldsymbol{D}}_n^\top \bar{\boldsymbol{D}} \boldsymbol{H} + \ell_n}, \tag{7}$$

$$\boldsymbol{D}_s \leftarrow \bar{\boldsymbol{D}}_s \bullet \frac{\boldsymbol{X}\boldsymbol{H}_s^\top + \bar{\boldsymbol{D}}_s \bullet \big(\boldsymbol{1}\big(\bar{\boldsymbol{D}}\boldsymbol{H}\boldsymbol{H}_s^\top \bullet \bar{\boldsymbol{D}}_s\big)\big)}{\bar{\boldsymbol{D}}\boldsymbol{H}\boldsymbol{H}_s^\top + \bar{\boldsymbol{D}}_s \bullet \big(\boldsymbol{1}\big(\boldsymbol{X}\boldsymbol{H}_s^\top \bullet \bar{\boldsymbol{D}}_s\big)\big)}. \tag{8}$$

We have introduced different sparsity parameters for the speech and noise because we hypothesize that having different sparsity for the speech and noise can improve the performance of the algorithm.

To reduce the wind noise in a recording we first compute the NNSC decomposition of an isolated recording of the wind noise using Equation (4–5). We discard the code matrix and use the noise dictionary matrix to compute the NNSC decomposition of the noisy signal using Equation (7–8). Finally we estimate the clean speech as

$$\hat{\boldsymbol{X}}_s = \bar{\boldsymbol{D}}_s \boldsymbol{H}_s. \tag{9}$$

To compute the waveform of the processed signal, we invert the STFT using the phase of the noisy signal.

3 Experimental results

To evaluate the algorithm we first used a test set consisting of eight phonetically diverse sentences from the Timit database. The sentences were spoken by different speakers, half of each gender. The speech signals were normalized to unit variance. We recorded wind noise outdoors using a setup emulating the microphone and amplifier in a hearing aid. We used half a minute of wind noise for estimating the noise dictionary. The signals were sampled at 16 kHz and the STFT were computed with a 32 ms Hanning window and 75% overlap. We mixed speech and wind noise at signal-to-noise ratios (SNR) of 0, 3, and 6 dB. In all our experiments the stopping criterion for the algorithm was when the relative change in the squared error was less than 10^{-4} or at a maximum of 500 iterations. As for most non-negative matrix factorization methods, the NNSC algorithm is prone to finding local minima and thus a suitable multi-start or multi-layer approach could be used [13]. In practice, however, we obtained good solutions using only a single run of the NNSC algorithm.

3.1 Initial setting of parameters

To find good initial values for the parameters of the algorithm, we evaluated the results on an empirically chosen range of values for each of the parameters shown below.

$\gamma \in \{.5, \underline{.6}, .7, .8\}$ The exponent of the short time Fourier tranform.

$\lambda_n \in \{\underline{.2}, .5\}$ The sparsity parameter used for learning the wind noise dictionary.

$N_s \in \{32, \underline{64}, 128\}$ The number of components in the speech dictionary.

$N_n \in \{4, 16, \underline{64}, 128\}$ The number of components in the wind noise dictionary.

$\ell_s \in \{\underline{.05}, .1, .2\}$ The sparsity parameter used for the speech code during separation.

$\ell_n \in \{\underline{0}, .1\}$ The sparsity parameter used for the noise code during separation.

For each of the 576 combinations of parameter settings, we computed the average increase in SNR. In total, more than six hours of audio was processed. The underlined parameter settings gave the highest increase in SNR. We used these parameter settings as a starting point for our furhter experiments. An example of the result of the algorithm is illustrated in Figure 2.

3.2 Importance and sensitivity of parameters

Next, we varied the parameters one by one while keeping the others fixed to the value chosen above. In these experiments, the input SNR was fixed at 3 dB. Figure 3–8 show the results; the box plots shows the median, upper and lower quartiles, and the range of the data. In the following we comment on each parameter in detail.

- γ (See Figure 3) The exponent of the STFT appears to be quite important. The best results in terms of SNR is achieved around $\gamma = 0.7$, although the algorithm is not particularly sensitive as long as γ is chosen around 0.5–1. Noticably, results are significantly worse when using the power spectrogram representation, $\gamma = 2$. The estimated value of the exponent corresponds to a cube root compression of the power spectrogram which curiously is an often used approximation to account for the nonlinear human perception of intensity.

- λ_n (See Figure 4) The sparsity parameter used in estimating the wind noise dictionary does not significantly influence the SNR . Qualitatively, however, there is a difference between low and high sparsity. Listening to the processed signals we found that with a less sparsified noise dictionary, the noise was well removed, but the speech was slightly distorted. With a more sparsified dictionary, there was more residual noise. Thus, this parameter can be used to make a tradeoff between residual noise and distortion.

- N_s (See Figure 5) The number of components in the speech dictionary is a very important parameter. Naturally, a reasonable number of components is needed in order to be able to model the speech adequately. Qualitatively, when using too few components, the result is a very clean signal consisting only of the most dominant speech sounds, most often the vowels. Interestingly though, having too many components also reduces the performance, since excess components can be used to model the noise. In this study we found that $N_s = 64$ components gave the best results, but we expect that it is dependent on the length of the recordings and the setting of the sparsity parameters etc.

- N_n (See Figure 6) The number of components in the wind noise dictionary is also important. Our results indicate that at least $N_n = 32$ components must be used and that the performance does not decrease when more components are used. Since the noise dictionary is estimated on an isolated recording of wind noise, all the elements in the dictionary will be tailored to fit the noise.

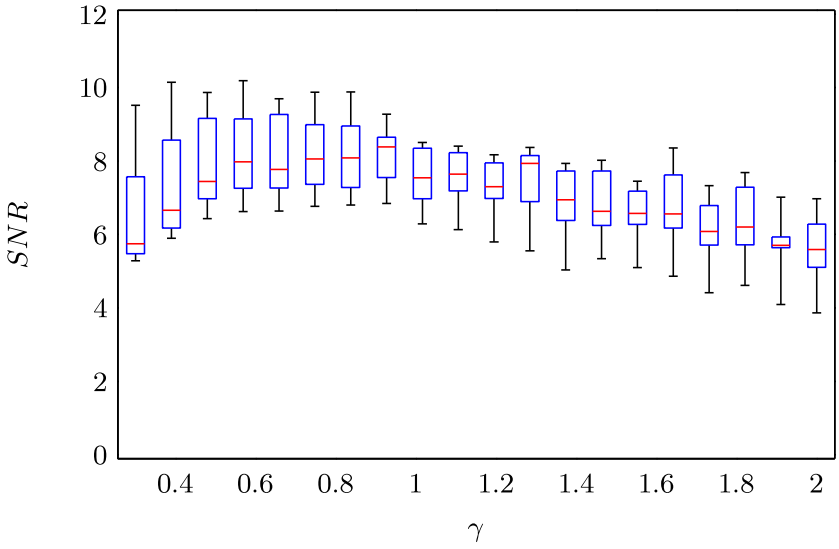

Figure 3: Exponent of the short time Fourier transform versus SNR. The best performance is achieved around $\gamma = 0.7$. The algorithm is not very sensitive to γ as long as it is chosen around 0.5–1.

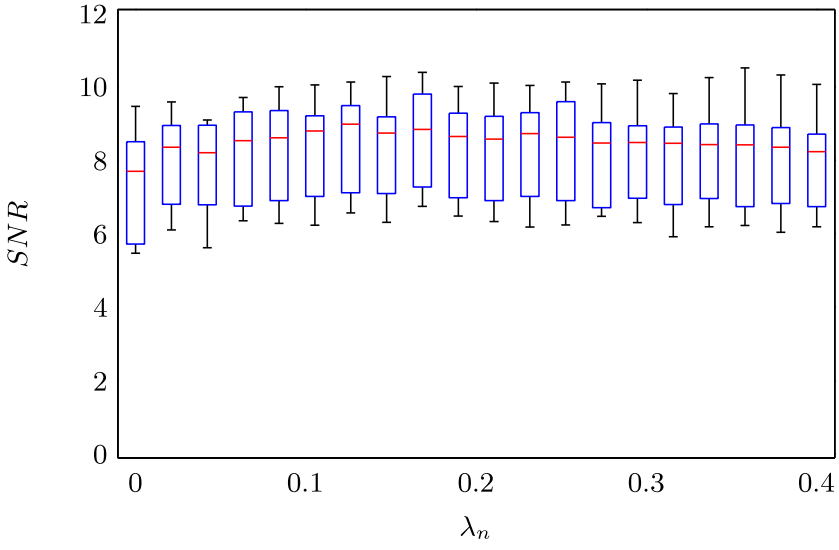

Figure 4: Sparsity parameter for the precomputation of the wind noise dictionary versus SNR. The method is not particularly sensitive to the selection of this parameter.

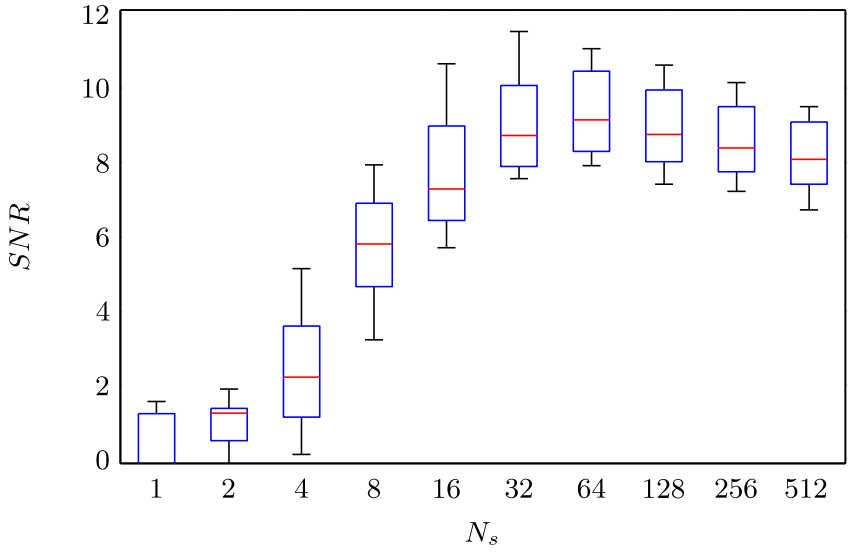

Figure 5: Number of components in the speech dictionary versus SNR. The best performance on the test set is achieved at $N_s = 64$. Using too few or too many components reduces the performance.

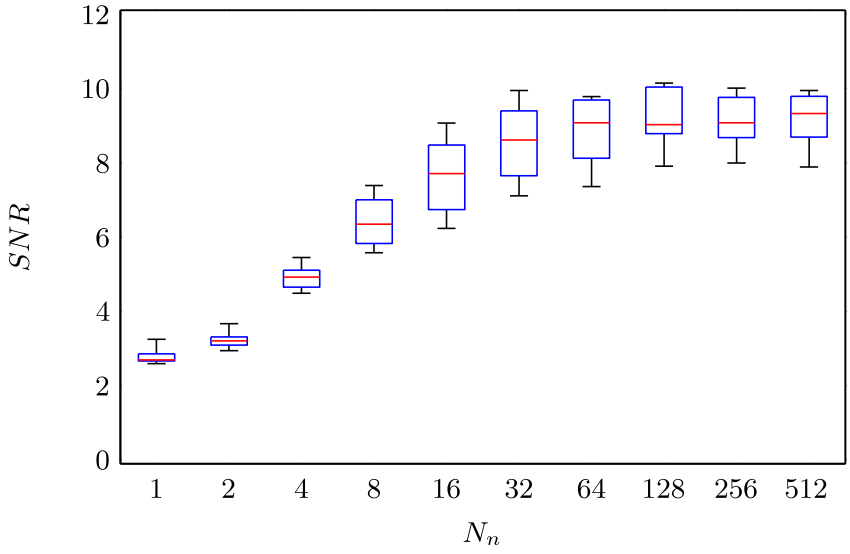

Figure 6: Number of components in the wind noise dictionary versus SNR. The results indicate that there should be at least $N_n = 32$ noise components.

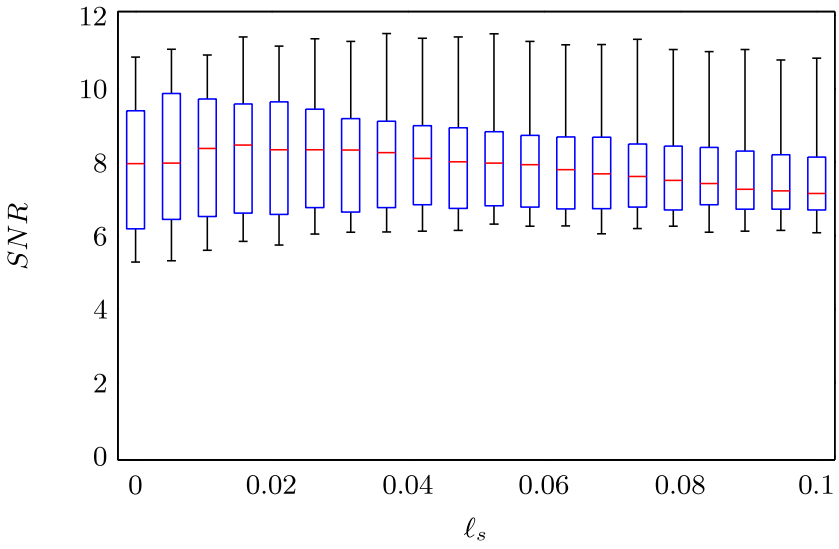

Figure 7: Sparsity parameter for the speech versus SNR. The method is not particularly sensitive to the selection of this parameter.

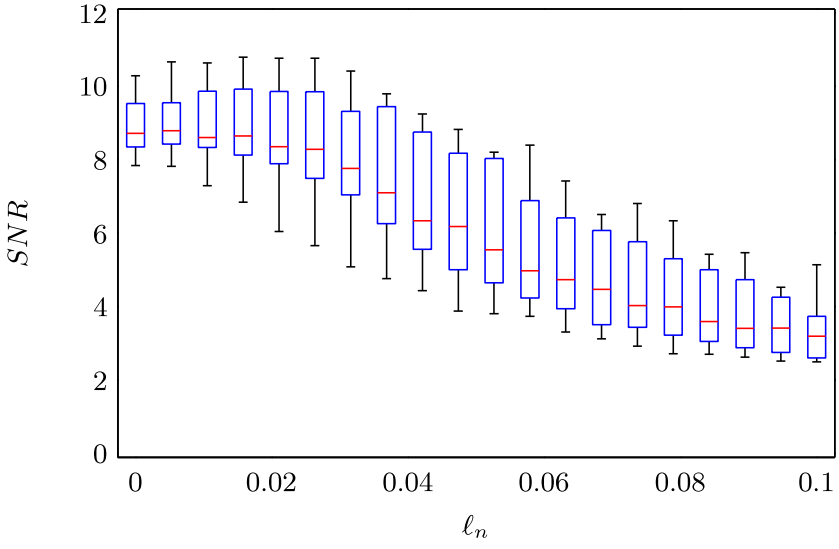

Figure 8: Sparsity parameter for the noise versus SNR. The method is very sensitive to the selection of this parameter, and it appears that no sparsity, $\ell_n = 0$, leads to the best performance.

ℓ_s (See Figure 7) The sparsity parameter used for the speech code does not appear very important when we look at the SNR, although slightly better results are obtained around $\ell_s = 0.02$. When we listen to the signals, however, there is a huge difference. When the parameter is close to zero, the noise in the processed signal is mainly residual wind noise. When the parameter is chosen in the high end of the range, there is not much wind noise left, but the speech is distorted. Thus, although not reflected in the SNR, this parameter balances residual noise and distortion similar to the sparsity parameter used for estimating the wind dictionary.

ℓ_n (See Figure 8) The sparsity parameter used for the wind noise during separation should basically be set to zero. Both qualitatively and in terms of SNR, imposing sparsity on the noise code only worsens performance. This makes sense, since the sparsity constrains the modeling ability of the noise dictionary, and consequently some of the noise is modeled by the speech dictionary.

3.3 Comparison with other methods

We compared our proposed metod for wind noise reduction to two other noise reduction methods. We used a test set consisting of 100 sentences from the GRID corpus. The sentences were spoken by a single female speaker. We mixed the speech with wind noise at different signal-to-noise ratios in the range 0–6 dB to see how the algorithm works under different noise conditions. All parameter settings were chosen as in the previous experiments.

We compared the results with the noise reduction in the Qualcomm-ICSI-OGI frontend for automatic speech recognition [15], which is based on adaptive Wiener filtering. We also compared to a simple spectral subtraction algorithm, implemented with an "oracle" voice activity detector. During non-speech activity we set the signal to zero and when speech was present we subtracted the spectrum of the noise taken from the last non-speech frame.

We computed two quality measures: i) the signal to noise ratio averaged over the 100 sentences and ii) the word recognition rate using an automatic speech recognition (ASR) system. The features used in the ASR were 13 Mel frequency cepstral coefficients plus Δ and $\Delta\Delta$ coefficients, and the system was based on a hidden Markov model with a 16 component Gaussian mixture model for each phoneme. The results are given in Figure 9– 10.

In terms of SNR, our proposed algorithm performs well (see Figure 9). The spectral subtraction algorithm also increases the SNR in all conditions, whereas the Qualcomm-ICSI-OGI algorithm actually decreases the SNR. In terms of word recognition rate the Qualcomm-ICSI-OGI algorithm gives the largest quality improvement (see Figure 10). This might not come as a surprise, since the algorithm is specifically designed for preprocessing in an ASR system. At low SNR, our proposed algorithm does increase the word recognition rate, but at high SNR, it is better not to use any noise reduction at all. The spectral sub-

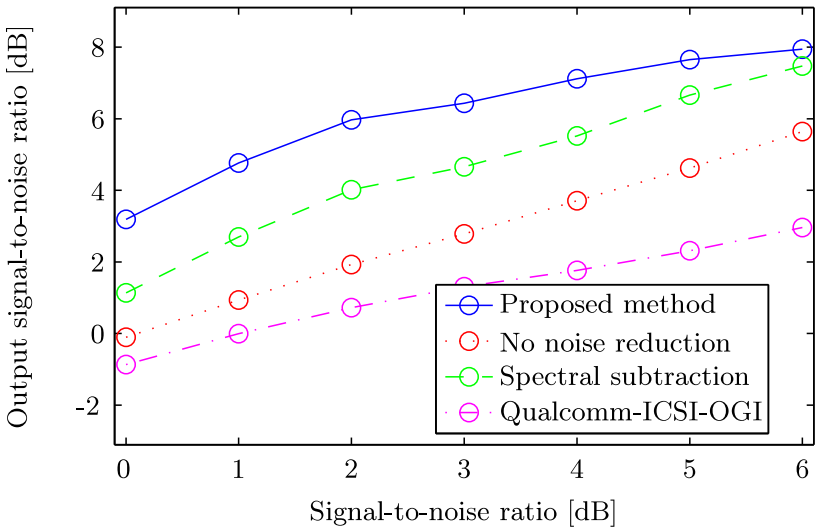

Figure 9: Output SNR versus input SNR. In terms of SNR, the proposed algorithm performs well.

traction algorithm performs much worse than using the original noisy speech in all conditions.

4 Discussion

We have presented an algorithm for reducing wind noise in recordings of speech based on estimating a source dictionary for the noise. The main idea was to make a system based on non-negative sparse coding, using a pre-estimated source model only for the noise. Our results show that the method is quite effective, and informal listening test indicate that often the algorithm is able to reduce sudden gusts of wind where other methods fail. In this work, we studied and optimized the performance in terms of signal-to-noise ratio, which is a simple but limited quality measure. Possibly, the algorithm will perform better in listening test and in speech recognition tasks, if the parameters are carefully tuned for these purposes, e.g., by optimizing a perceptual speech quality measure or word recognition rate.

References

[1] Steven F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *Acoustics, Speech, and Signal Processing, IEEE Transactions on*, vol. 27, no. 2, pp. 113–120, 1979.

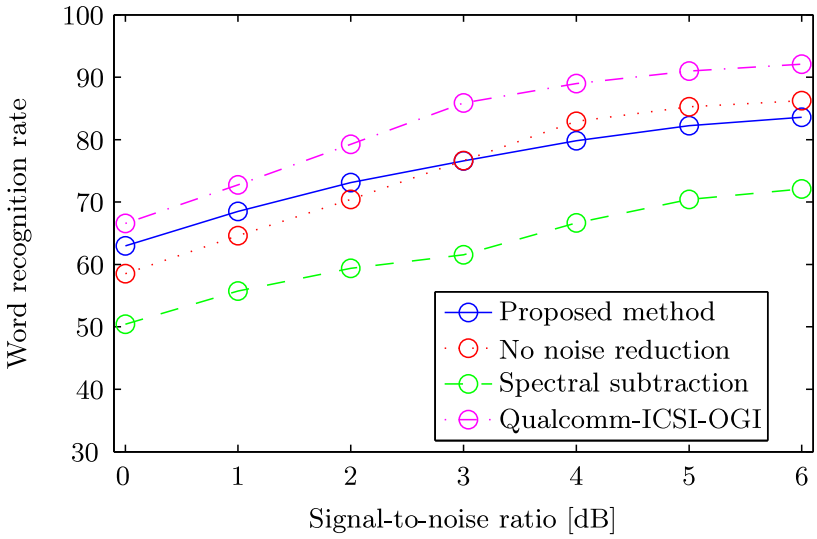

Figure 10: Word recognition rate on a speech recognition task versus input SNR. The Qualcomm-ICSI-OGI algorith which is designed for this purpose performs best. At low SNR our proposed algorithm gives better results than using the noisy speech directly.

[2] Sam T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems*, 2000, pp. 793–799.

[3] Sam T. Roweis, "Factorial models and refiltering for speech separation and denoising," in *Eurospeech*, 2003, pp. 1009–12.

[4] Daniel P. W. Ellis and Ron J. Weiss, "Model-based monaural source separation using a vector-quantized phase-vocoder representation," in *International Conference on Acoustics, Speech and Signal Processing*, may 2006, pp. 957–960.

[5] Mikkel N. Schmidt and Rasmus K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006.

[6] Ron J. Weiss and Daniel P. W. Ellis, "Estimating single-channel source separation masks: Relevance vector machine classifiers vs. pitch-based masking," in *Statistical and Perceptual Audio Processing, Workshop on*, 2006.

[7] M Berouti, R Schwartz, and J Makhoul, "Enhancement of speech corrupted by acoustic noise," in *International Conference on Acoustics, Speech and Signal Processing*, 1979, vol. 4, pp. 208–211.

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

[9] P.O. Hoyer, "Non-negative sparse coding," in *Neural Networks for Signal Processing, IEEE Workshop on*, 2002, pp. 557–565.

[10] Julian Eggert and Edgar Körner, "Sparse coding and NMF," in *Neural Networks, IEEE International Conference on*, 2004, vol. 4, pp. 2529–2533.

[11] Chih-Jen Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation (to appear)*, 2007.

[12] Dongmin Kim, Suvrit Sra, and Inderjit S. Dhillon, "Fast newton-type methods for the least squares nonnegative matrix approximation problem," in *Data Mining, Proceedings of SIAM Conference on*, 2007.

[13] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization," *Electronic Letters*, vol. 42, no. 16, pp. 947–958, 2006.

[14] B. Wang and M. D. Plumbley, "Musical audio stream separation by non-negative matrix factorization," in *DMRN Summer Conference, Glasgow, Proceedings of the*, july 2005.

[15] Andre Adami, Lukás Burget, Stephane Dupont, Hari Garudadri, Frantisek Grezl, Hynek Hermansky, Pratibha Jain, Sachin Kajarekar, Nelson Morgan, and Sunil Sivadas, "Qualcomm-icsi-ogi features for asr," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2002, pp. 21–24.

Paper D

Linear Regression on Sparse Features for Single-Channel Speech Separation

Mikkel N. Schmidt and Rasmus K. Olsson, "Linear Regression on Sparse Features for Single-Channel Speech Separation," in *Applications of Signal Processing to Audio and Acoustics, IEEE Workshop on (WASPAA)*, Oct. 2007.

Linear Regression on Sparse Features for Single-Channel Speech Separation

Mikkel N. Schmidt and Rasmus K. Olsson
Technical University of Denmark
Richard Petersens Plads, Bldg. 321
DK-2800 Kgs. Lyngby, Denmark
Email: {mns,rko}@imm.dtu.dk

Abstract

In this work we address the problem of separating multiple speakers from a single microphone recording. We formulate a linear regression model for estimating each speaker based on features derived from the mixture. The employed feature representation is a sparse, non-negative encoding of the speech mixture in terms of pre-learned speaker-dependent dictionaries. Previous work has shown that this feature representation by itself provides some degree of separation. We show that the performance is significantly improved when regression analysis is performed on the sparse, non-negative features, both compared to linear regression on spectral features and compared to separation based directly on the non-negative sparse features.

1 Introduction

The cocktail-party problem can be defined as that of isolating or recognizing speech from an individual speaker in the presence of interfering speakers. The ability of the human auditory system to solve this problem is impressive, even when using only one ear, or equivalently, listening to a mono recording of a mixture of different speakers. It is an interesting and currently unsolved research problem to devise an algorithm which can mimic this ability.

Different approaches for constructing such a system have been proposed, including methods based on computational auditory scene analysis (CASA) inspired by the mechanisms of the human auditory system; blind source separation (BSS) using little or no prior information about the signals; and machine learning methods, where speech models are learned from training data and subsequently used to separate the mixed speech. In this paper we focus on the machine learning approach, where isolated recordings of the individual speakers we wish to separate are available for training.

A number of such methods have been propsed. One approach, which arguably has been the most successful, is to use a hidden Markov model (HMM) based on a Gaussian mixture model (GMM) for each speech source and combine these in a factorial HMM to separate a mixture [1]. Direct inference in such a model is not practical because of the dimensionality of the combined state space of the factorial HMM. Roweis [1] shows how to obtain tractable inference by exploiting the fact that in a log-magnitude time-frequency representation, the sum of speech signals is well approximated by the maximum. Recently, impressive results have been achieved by Kristjansson et al. [2] who have devised an efficient method of inference that does not use the max-approximation. In some situations, their system exceeds human performance in terms of the error rate in a word recognition task.

Another class of algorithms, here denoted 'dictionary methods', generally rely on learning a matrix factorization, in terms of a dictionary and its encoding for each speaker, from training data. The dictionary is a source dependent basis, and the method relies on the dictionaries of the sources in the mixture being sufficiently different. Separation of a mixture is obtained by computing the combined encoding using the concatenation of the source dictionaries. As opposed to the HMM/GMM based methods, this does not require a combinatorial search and leads to faster inference. Different matrix factorization methods can be conceived based on various a priori assumptions. For instance, independent component analysis and sparse decomposition, where the encoding is assumed to be sparsely distributed, have been proposed for single-channel speech separation [3, 4]. Another way to constrain the matrices is achieved through the assumption of non-negativity [5, 6], which is especially relevant when modeling speech in a magnitude spectrogram representation. Sparsity and non-negativity priors have been combined in sparse, non-negative matrix factorization [7] and applied to music and speech separation tasks [8, 9, 10].

In this work, we formulate a linear regression model for separating a mixture of speech signals based on features derived from a time-frequency representation of the speech. As a set of features, we use the encodings pertaining to dictionaries learned for each speaker using sparse, non-negative matrix factorization. We evaluate the performance of the method on synthetic speech mixtures by computing the signal-to-error ratio, which is the simplest, arguably sufficient, quality measure [11].

2 Methodology

The problem is to estimate P speech sources from a single microphone recording,

$$y(t) = \sum_{i=1}^{P} y_i(t), \tag{1}$$

where $y(t)$ and $y_i(t)$ are the time-domain mixture and source signals respectively.

We compute the separation in a time-frequency magnitude representation, $\boldsymbol{Y} = \mathrm{TF}\{y(t)\}$, where $\boldsymbol{Y}$ is a non-negative real-valued matrix with spectral

vectors as columns, i.e., we do not try to estimate the phase. Instead, to compute the separated time-domain signals, we refilter the original mixture signal using the estimated magnitude spectra.

2.1 Linear regression

To perform the separation we propose a simple method, namely linear regression. We estimate the magnitude time-frequency representations of the sources in a mixture as a linear regression on features derived from the mixture. The linear model reads,

$$\boldsymbol{Y}_i = \boldsymbol{W}_i^\top (\boldsymbol{X} - \boldsymbol{\mu}\mathbf{1}^\top) + \boldsymbol{m}_i\mathbf{1}^\top + \boldsymbol{N}, \tag{2}$$

where $\boldsymbol{Y}_i = \mathrm{TF}\{y_i(t)\}$ is the time-frequency representation of the i'th source, $\boldsymbol{W}_i$ is a matrix of weights, $\boldsymbol{X}$ is a feature matrix derived from $\boldsymbol{Y}$; in the following we discuss these features in detail. The vectors $\boldsymbol{\mu}$ and $\boldsymbol{m}_i$ are the means of the features and the sources respectively and are computed on training data. The matrix $\boldsymbol{N}$ is an additive noise term.

We make two assumptions in order to obtain a particularly simple maximum a posteriori (MAP) estimator based on this model: i) the noise is zero mean normal i.i.d. with variance σ_n^2 and ii) the prior distribution of the weights is zero mean normal i.i.d. with variance σ_w^2. For a detailed derivation of the MAP estimator, see e.g. Rasmussen and Williams [12]. Under these assumptions, the MAP estimator of the i'th source is given by

$$\hat{\boldsymbol{Y}}_i^* = \boldsymbol{\Gamma}_i\boldsymbol{\Sigma}^{-1}(\boldsymbol{X}^* - \boldsymbol{\mu}\mathbf{1}^\top) + \boldsymbol{m}_i\mathbf{1}^\top, \tag{3}$$

where $\boldsymbol{X}^*$ is the feature matrix computed from the test mixture, $\boldsymbol{Y}^*$, and

$$\boldsymbol{\Gamma}_i = \big(\boldsymbol{Y}_i - \boldsymbol{m}_i\mathbf{1}^\top\big)\big(\boldsymbol{X} - \boldsymbol{\mu}\mathbf{1}^\top\big)^\top, \tag{4}$$

$$\boldsymbol{\Sigma} = \big(\boldsymbol{X} - \boldsymbol{\mu}\mathbf{1}^\top\big)\big(\boldsymbol{X} - \boldsymbol{\mu}\mathbf{1}^\top\big)^\top + \frac{\sigma_n^2}{\sigma_w^2}\boldsymbol{I}. \tag{5}$$

Here, $\boldsymbol{X}$ is a matrix with feature vectors computed on a training mixture and $\boldsymbol{Y}_i$ is the corresponding time-frequency representation of the source.

When an isolated recording, $\boldsymbol{Y}_i$ is available as training data for each of the speakers, it is necessary to construct the training feature matrix, $\boldsymbol{X}$, from synthetic mixtures. One way to exploit the available data would be to generate mixtures, $\boldsymbol{X}$, such that all possible combinations of time-indices are represented. However, the number of sources and/or the number of available time-frames would be prohibitively large. For example, the five minute training data used for each speaker in this paper lead to matrices $\boldsymbol{Y}_i$ with approximately 10^4 columns. Creating all combinations of just two speakers would require computing a feature matrix, $\boldsymbol{X}$, having 10^8 columns.

A feasible approximation can be found in the limit of a large training set by making two additional assumptions: i) the features are additive, $\boldsymbol{X} = \sum_i^P \boldsymbol{X}_i$ with mean vectors $\boldsymbol{\mu}_i$, which is reasonable for, e.g., sparse features, and ii)

the features are uncorrelated between sources such that all cross-products are negligible. Then, we can make the following approximation

$$\boldsymbol{\Gamma}_i \approx \left(\boldsymbol{Y}_i - \boldsymbol{m}_i\mathbf{1}^\top\right)\left(\boldsymbol{X_i} - \boldsymbol{\mu}_i\mathbf{1}^\top\right)^\top, \tag{6}$$

$$\boldsymbol{\Sigma} \approx \sum_{i=1}^{P}\left(\boldsymbol{X}_i - \boldsymbol{\mu}_i\mathbf{1}^\top\right)\left(\boldsymbol{X}_i - \boldsymbol{\mu}_i\mathbf{1}^\top\right)^\top, \tag{7}$$

which allows us to use isolated recordings of each source as training data directly without generating synthetic mixtures.

2.2 Features

In this work, we explore two sets of feature mappings. The first, and most simple, is to use the mixture time-frequency representation itself as input to the linear model, $\boldsymbol{X}_i = \boldsymbol{Y}_i$, $\boldsymbol{X}^* = \boldsymbol{Y}^*$. With these features, the spectra of each speaker is modeled as a linear combination of the mixed speech spectra; this allows the model to capture correlations between frequency bands specific to each speaker.

The second feature set we explore is the encodings of a sparse, non-negative matrix factorization algorithm (SNMF) [7]. Possibly, other dictionary methods provide equally viable features. In the SNMF method, the time-frequency representation of the i'th source is modelled as $\boldsymbol{Y}_i \approx \boldsymbol{D}_i\boldsymbol{H}_i$ where $\boldsymbol{D}_i$ is a dictionary matrix containing a set of spectral basis vectors, and $\boldsymbol{H}_i$ is an encoding which describes the amplitude of each basis vector at each time point. In order to use the method to compute features for a mixture, a dictionary matrix is first learned separately on a training set for each of the sources. Next, the mixture and the training data is mapped onto the concatenated dictionaries of the sources,

$$\boldsymbol{Y}_i \approx \boldsymbol{D}_i\boldsymbol{H}_i, \qquad \boldsymbol{Y}^* \approx \boldsymbol{D}\boldsymbol{H}^*, \tag{8}$$

where $\boldsymbol{D} = [\boldsymbol{D}_1, \ldots, \boldsymbol{D}_P]$. The encoding matrices, $\boldsymbol{H}_i$ and $\boldsymbol{H}^*$, are then the features used as input to the linear model, $\boldsymbol{X}_i = \boldsymbol{H}_i$, $\boldsymbol{X}^* = \boldsymbol{H}^*$.

In previous work, the sources were estimated directly from these features as $\hat{\boldsymbol{Y}}_i^* = \boldsymbol{D}_i\boldsymbol{H}_i^*$ [10]. For comparison, we include this method in our evaluations. This method yields very good results when the sources, and thus the dictionaries, are sufficiently different from each other. In practice, however, this will not always be the case. In the factorization of the mixture, $\boldsymbol{D}_1$ may not only encode $\boldsymbol{Y}_1$ but also $\boldsymbol{Y}_2$ etc. This suggests that the encodings should rather be used as features in an estimator for each source.

3 Evaluation

The proposed speech separation method was evaluated on a subset of the GRID speech corpus [13] consisting of the first 4 male and first 4 female speakers (no. 1, 2, 3, 4, 5, 7, 11, and 15). The data was preprocessed by concatenating 5

minutes of speech from each speaker and resampling to 8 kHz. As a measure of performance, the signal-to-error ratio (SER) averaged across sources was computed in the time-domain. The testing was performed on a total of 9 minutes of synthetic 0 dB mixtures of two speakers, constructed using all combinations of speakers in the test set.

The time-frequency representation of the sources and mixtures were computed by normalizing the time-signals to unit power and computing the short-time Fourier transform (STFT) using 64 ms Hamming windows with 50% overlap. The absolute value of the STFT was then mapped onto a mel frequency scale using a publicly available toolbox [14] in order to reduce the dimensionality. Finally, the mel-frequency magnitude spectrogram was amplitude-compressed by exponentiating to the power p. By cross-validation we found that best results were obtained at $p = 0.55$ which gave significantly better results compared with, e.g., operating in the amplitude ($p = 1$) or the power ($p = 2$) domains (see Figure 4). Curiously, this is similar to the empirically determined $p \approx 0.67$ exponent used in power law modelling of perceived loudness in humans, known as Stevens' Law (see for example Hermansky [15]).

When learning the sparse dictionaries, the SNMF algorithm was allowed 250 iterations to converge from random initial conditions drawn from a uniform distribution on the unit interval. The number of dictionary atoms was fixed at 200. The SNMF method has a sparsity parameter, λ, which we chose by cross-validation to $\lambda = 0.15$. When computing the encodings on the test mixtures, we did not enforce sparsity, as cross-validation showed that best results were obtained at $\lambda = 0$.

Since the methods separate speakers in the magnitude time-frequency domain and do not estimate the phase of the separated signals, we used a simple refiltering method to compute separated time-domain signals. We computed the STFT of the mixture signal and performed a binary masking and subsequent inversion as described by Wang and Brown [16]. Audio examples of the reconstructed speech are available online [17].

In Figures 1 and 2, the performance is shown for the different methods. The acronyms MAP-Mel and MAP-SNMF refer to using the mel spectrum or the SNMF encoding as features in the linear regression, respectively. For reference, results are provided for the basic SNMF approach as well [10].

We also experimented with using a stacked feature representation, where five consecutive feature vectors spaced 32 ms apart were combined into one large feature vector as a simple means to modeling temporal dynamics. In the figures, this is denoted by the suffix "5".

The best performance is achieved for MAP-SNMF-5, reaching an $\simeq 1.2$ dB average improvement over the SNMF algorithm. It is noteworthy that the improvement is larger for the most difficult mixtures, those involving same-gender speakers.

In order to verify that the method is robust to changes in the relative gain of the signals in the mixtures, the performance was evaluated in a range of different target-to-interference ratios (TIR) (see Figure 3). The results indicate that the method works very well even when the TIR is not known a priori.

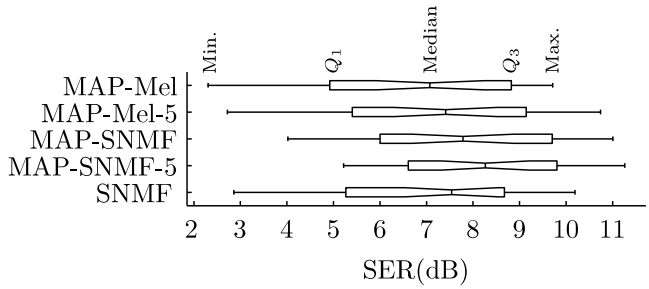

Figure 1: The distribution of the signal-to-error (SER) performance of the method for all combinations of two speakers. The mel magnitude spectrogram (MAP-Mel) and the SNMF encodings (MAP-SNMF) were used as features to the linear model. The results of using basic SNMF are given as a reference. The box plots indicate the extreme values along with the quartiles of the dB SER, averaged across sources.

In Figure 5, the performance is measured as a function of the available training data, indicating that the method is almost converged when using 5 minutes of training data.

4 Discussion

The main idea in this paper was to use sparse coding features in a linear estimation scheme. We have shown that this approach leads to better performance compared to linear regression on spectral features and compared to separation using the sparse features directly. Our results warrant further studies of the use of sparse features for speech separation, possibly using a more sophisticated estimator than the linear regression model discussed here.

The computation in the linear model is fast, since the estimation of the separation matrix is closed-form given the features. The SNMF for computing the dictionaries and the sparse feature mapping of the mixture, however, is quite expensive. A possible remedy for the latter computations could be to devise a greedy approximation.

We experimented with concatenating features across time as a simple means of modeling the temporal dynamics of speech. Doing this appears to improve performance slightly, but the effect is relatively small, confirming previous reports that the inclusion of an acoustical dynamical model yields only marginal improvements [2], [18].

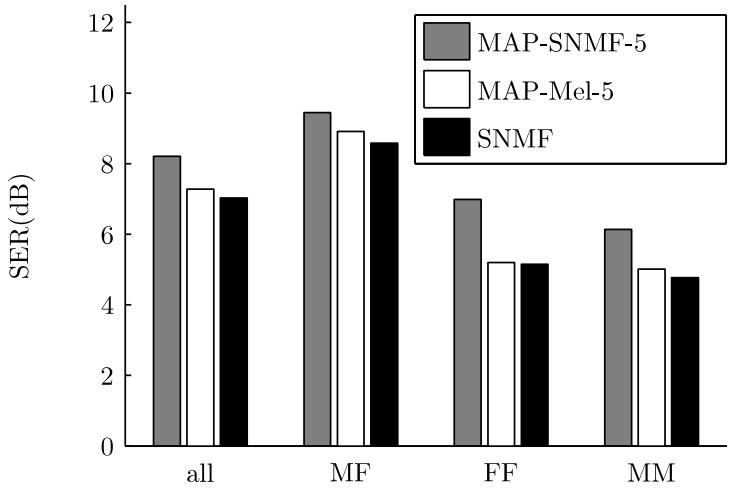

Figure 2: The performance of the methods given as signal-to-error (SER) in dB, depending on the gender of the speakers. Male and female are identified by 'M' and 'F', respectively. The improvement of MAP-SNMF-5 over MAP-Mel-5 and SNMF is largest in the most difficult (same-gender) mixtures.

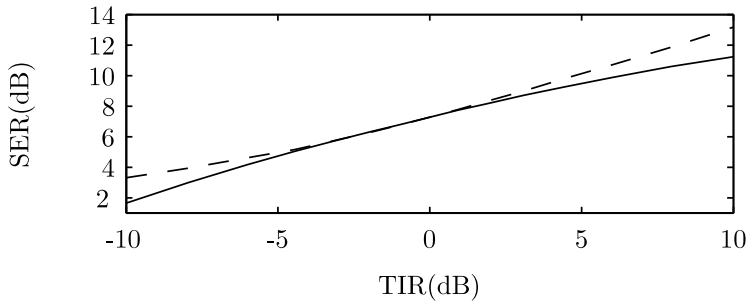

Figure 3: The performance of the MAP-Mel-5 algorithm given as the signal-to-error ratio (SER) of the target signal versus the target-to-interference ratio (TIR) of the mixture. The solid and dashed curves represent training on 0dB or the actual TIR of the test mixture, respectively. Clearly, the method is robust to a mismatch of the TIR between the training and test sets.

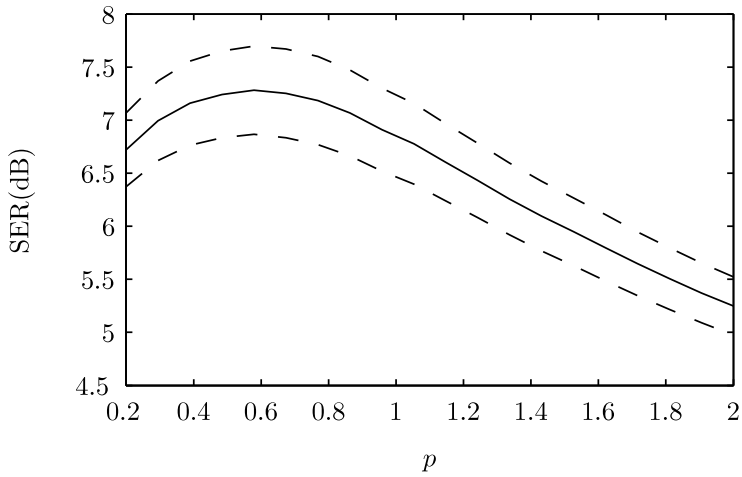

Figure 4: The effect of amplitude compression on the performance of the MAP-Mel-5 algorithm as measured in the signal-to-error ratio (SER). The optimal value of the exponent was found at $p \simeq 0.55$, in approximate accordance with Steven's power law for hearing. The dashed curve indicates the standard deviation of the mean.

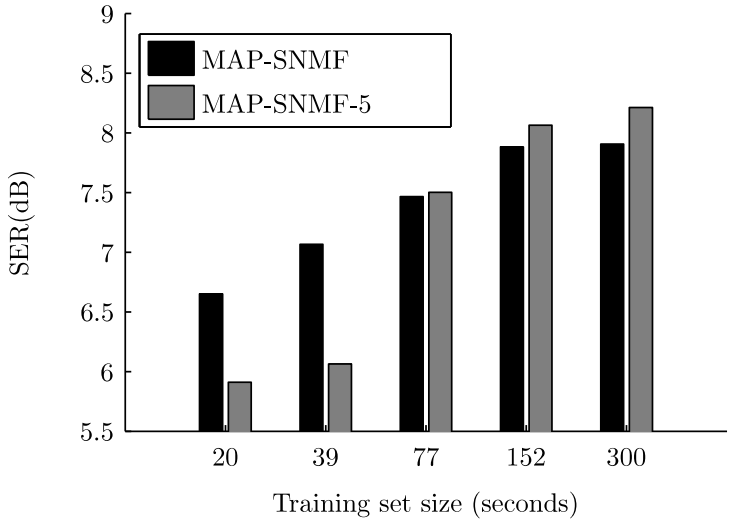

Figure 5: The learning curve of the method, measured in signal-to-error ratio (SER), as a function of the size of the training set, depending on the complexity of the method.

5 Acknowledgment

During the research process, L. K. Hansen, J. Larsen and O. Winther administered advice and good suggestions.

References

[1] S. T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems*, 2000, pp. 793–799.

[2] T. Kristjansson, J. Hershey, P. Olsen, S. Rennie, and R. Gopinath, "Super-human multi-talker speech recognition: The IBM 2006 speech separation challenge system," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006, pp. 97–100.

[3] G. J. Jang and T. W. Lee, "A maximum likelihood approach to single channel source separation," *Journal of Machine Learning Research*, vol. 4, pp. 1365–1392, 2003.

[4] B. A. Pearlmutter and R. K. Olsson, "Algorithmic differentiation of linear programs for single-channel source separation," in *Machine Learning and Signal Processing, IEEE International Workshop on*, 2006.

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

[6] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," in *Statistical and Perceptual Audio Processing (SAPA)*, 2004.

[7] J. Eggert and E. Körner, "Sparse coding and NMF," in *Neural Networks, IEEE International Conference on*, vol. 4, 2004, pp. 2529–2533.

[8] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *International Computer Music Conference, ICMC*, 2003.

[9] L. Benaroya, L. M. Donagh, F. Bimbot, and R. Gribonval, "Non negative sparse representation for wiener based source separation with a single sensor," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, 2003, pp. 613–616.

[10] M. N. Schmidt and R. K. Olsson, "Single-channel speech separation using sparse non-negative matrix factorization," in *International Conference on Spoken Language Processing (INTERSPEECH)*, 2006.

[11] D. Ellis, "Evaluating speech separation systems," in *Speech Separation by Humans and Machines*, P. Divenyi, Ed. Kluwer Academic Publishers, ch. 20, pp. 295–304.

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[13] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *submitted to JASA*.

[14] D. P. W. Ellis. (2005) PLP and RASTA (and MFCC, and inversion) in Matlab. [Online]. Available: http://www.ee.columbia.edu/∼dpwe/resources/matlab/rastamat

[15] H. Hermansky, "Perceptual linear predictive (plp) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[16] D. L. Wang and G. J. Brown, "Separation of speech from interfering sounds based on oscillatory correlation," *IEEE-NN*, vol. 10, no. 3, p. 684, 1999.

[17] M. N. Schmidt and R. K. Olsson. (2007) Audio samples relevant to this paper. [Online]. Available: http://mikkelschmidt.dk/waspaa2007

[18] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transaction on Audio, Speech and Language Processing - to appear*, 2007.

Paper E

Non-negative Matrix Factorization with Gaussian Process Priors

Mikkel N. Schmidt and Hans Laurberg, "Non-negative Matrix Factorization with Gaussian Process Priors," in *Computational Intelligence and Neuroscience*, May 2008.

Non-negative Matrix Factorization with Gaussian Process Priors

Mikkel N. Schmidt
Technical University of Denmark
Informatics and Mathematical Modelling
Richard Petersens Plads, Building 321
2800 Kgs. Lyngby, Denmark
mns@imm.dtu.dk

Hans Laurberg
Aalborg University
Department of Electronic Systems
Niels Jernes Vej 12
9220 Aalborg Ø., Denmark
hla@es.aau.dk

Abstract

We present a general method for including prior knowledge in a non-negative matrix factorization (NMF), based on Gaussian process priors. We assume, that the non-negative factors in the NMF are linked by a strictly increasing function to an underlying Gaussian process, specified by its covariance function. This allows us to find NMF decompositions, that agree with our prior knowledge of the distribution of the factors, such as sparseness, smoothness, and symmetries. The method is demonstrated with an example from chemical shift brain imaging.

1 Introduction

Non-negative matrix factorization (NMF) [1, 2] is a recent method for factorizing a matrix as the product of two matrices, in which all elements are non-negative. NMF has found widespread application in many different areas including pattern recognition [3], clustering [4], dimensionality reduction [5], and spectral analysis [6, 7]. Many physical signals, such as pixel intensities, amplitude spectra, and occurence counts, are naturally represented by non-negative numbers. In the analysis of mixtures of such data, non-negativity of the individual components is a reasonable constraint. Recently, a very simple algorithm [8] for computing the NMF was introduced. This has initiated much research aimed at developing more robust and efficient algorithms.

Efforts have been made to enhance the quality of the NMF by adding further constraints to the decomposition, such as sparsity [9], spatial localization [10, 11], and smoothness [11, 12], or by extending the model to be convolutive [13, 14]. Many extended NMF methods are derived by adding appropriate constraints and penalty terms to a cost function. Alternatively, NMF methods can be derived in a probabilistic setting, based on the distribution of the data [15, 16, 6, 17]. These approaches have the advantage that the underlying assumptions in the model are made explicit.

In this paper we present a general method for using prior knowledge to improve the quality of the solutions in NMF. The method is derived in a probabilistic setting, and it is based on defining prior probability distributions of the factors in the NMF model in a Gaussian process framework. We assume that the non-negative factors in the NMF are linked by a strictly increasing function to an underlying Gaussian process, specified by its covariance function. By specifying the covariance of the underlying process, we can compute NMF decompositions that agree with our prior knowledge of the factors, such as sparseness, smoothness, and symmetries. We refer to the proposed method as non-negative matrix factorization with Gaussian process priors, or GPP-NMF for short.

2 NMF with Gaussian Process Priors

In the following we derive a method for including prior information in an NMF decomposition by assuming Gaussian process priors (for a general introduction to Gaussian processes, see e.g. Rasmussen and Williams [18].) In our approach, the Gaussian process priors are linked to the non-negative factors in the NMF by a suitable link function. To set up the notation, we start by deriving the standard NMF method as a maximum likelihood (ML) estimator and then move on to the maximum a posteriori (MAP) estimator. Then we discuss Gaussian process priors and introduce a change of variable that gives better optimization properties. Finally, we discuss the selection of the link function.

2.1 Maximum Likelihood NMF

The NMF problem can be stated as

$$\boldsymbol{X} = \boldsymbol{D}\boldsymbol{H} + \boldsymbol{N}, \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{K \times L}$ is a data matrix that is factorized as the product of two element-wise non-negative matrices, $\boldsymbol{D} \in \mathbb{R}_+^{K \times M}$ and $\boldsymbol{H} \in \mathbb{R}_+^{M \times L}$, where $\mathbb{R}_+$ denotes the non-negative reals. The matrix $\boldsymbol{N} \in \mathbb{R}^{K \times L}$ is the residual noise.

There exists a number of different algorithms [8, 19, 20, 21, 16, 15, 17] for computing this factorization, some of which can be viewed as maximum likelihood methods under certain assumptions about the distribution of the data. For example, least squares NMF corresponds to i.i.d. Gaussian noise [17] and Kullback-Leibler NMF corresponds to a Poisson process [6].

The ML estimate of $\boldsymbol{D}$ and $\boldsymbol{H}$ is given by

$$\{\boldsymbol{D}_{\mathrm{ML}}, \boldsymbol{H}_{\mathrm{ML}}\} = \underset{\boldsymbol{D},\boldsymbol{H}\geq 0}{\arg\min}\, \mathcal{L}_{X|D,H}(\boldsymbol{D}, \boldsymbol{H}), \tag{2}$$

where $\mathcal{L}_{X|D,H}(\boldsymbol{D}, \boldsymbol{H})$ is the negative log likelihood of the factors.

Example 1 (Least squares NMF)**.** *An example of a maximum likelihood NMF is the least squares estimate. If the noise is i.i.d. Gaussian with variance* σ_N^2*, the likelihood of the factors* $\boldsymbol{D}$ *and* $\boldsymbol{H}$ *can be written as*

$$p^{\mathrm{LS}}_{X|D,H}(\boldsymbol{X}|\boldsymbol{D}, \boldsymbol{H}) = \frac{1}{\left(\sqrt{2\pi}\sigma_N\right)^{KL}} \exp\left(-\frac{||\boldsymbol{X} - \boldsymbol{D}\boldsymbol{H}||_F^2}{2\sigma_N^2}\right). \tag{3}$$

The negative log likelihood, which serves as a cost function for optimization, is then

$$\mathcal{L}^{\mathrm{LS}}_{X|D,H}(\boldsymbol{D}, \boldsymbol{H}) \propto \frac{1}{2\sigma_N^2}||\boldsymbol{X} - \boldsymbol{D}\boldsymbol{H}||_F^2, \tag{4}$$

where we use the proportionality symbol to denote equality subject to an additive constant. To compute a maximum likelihood estimate of $\boldsymbol{D}$ *and* $\boldsymbol{H}$*, the gradient of the negative log likelihood is useful*

$$\nabla_H \mathcal{L}^{\mathrm{LS}}_{X|D,H}(\boldsymbol{D}, \boldsymbol{H}) = \frac{1}{\sigma_N^2}\boldsymbol{D}^\top(\boldsymbol{D}\boldsymbol{H} - \boldsymbol{X}), \tag{5}$$

and the gradient with respect to $\boldsymbol{D}$*, which is easy to derive, is similar because of the symmetry of the NMF problem.* □

The ML estimate can be computed by multiplicative update rules based on the gradient [8], projected gradient descent [19], alternating least squares [20], Newton-type methods [21], or any other appropriate constrained optimization method.

2.2 Maximum a Posteriori NMF

In this paper, we propose a method to build prior knowledge into the solution of the NMF problem. We choose a prior distribution $\mathrm{p}_{D,H}(\boldsymbol{D}, \boldsymbol{H})$ over the factors in the model, that captures our prior beliefs and uncertainties of the solution we seek. We then compute the maximum a posteriori (MAP) estimate of the factors. Using Bayes rule, the posterior is given by

$$\mathrm{p}_{D,H|X}(\boldsymbol{D}, \boldsymbol{H}|\boldsymbol{X}) = \frac{\mathrm{p}_{X|D,H}(\boldsymbol{X}|\boldsymbol{D}, \boldsymbol{H})\mathrm{p}_{D,H}(\boldsymbol{D}, \boldsymbol{H})}{\mathrm{p}_X(\boldsymbol{X})}. \tag{6}$$

Since the numerator is constant, the negative log posterior is the sum of a likelihood term that penalizes model misfit, and a prior term that penalizes solutions that are unlikely under the prior

$$\mathcal{L}_{D,H|X}(\boldsymbol{D}, \boldsymbol{H}) \propto \mathcal{L}_{X|D,H}(\boldsymbol{D}, \boldsymbol{H}) + \mathcal{L}_{D,H}(\boldsymbol{D}, \boldsymbol{H}). \tag{7}$$

The MAP estimate of $\boldsymbol{D}$ and $\boldsymbol{H}$ is

$$\{\boldsymbol{D}_{\text{MAP}}, \boldsymbol{H}_{\text{MAP}}\} = \arg\min_{\boldsymbol{D},\boldsymbol{H}\geq 0} \mathcal{L}_{D,H|X}(\boldsymbol{D}, \boldsymbol{H}), \tag{8}$$

and it can again be computed using any appropriate optimization algorithm.

Example 2 (Non-negative sparse coding)**.** *An example of a MAP NMF is non-negative sparse coding (NNSC) [9, 22], where the prior on $\boldsymbol{H}$ is i.i.d. exponential, and the prior on $\boldsymbol{D}$ is flat with each column constrained to have unit norm*

$$p_{D,H}^{\text{NNSC}}(\boldsymbol{D}, \boldsymbol{H}) = \prod_{i,j} \lambda \exp\left(-\lambda \boldsymbol{H}_{i,j}\right), \quad ||\boldsymbol{D}_k|| = 1 \ \forall k, \tag{9}$$

where $||\boldsymbol{D}_k||$ is the Euclidean norm of the k'th column of $\boldsymbol{D}$. This corresponds to the following penalty term in the cost function

$$\mathcal{L}_{D,H}^{\text{NNSC}}(\boldsymbol{D}, \boldsymbol{H}) \propto \lambda \sum_{i,j} \boldsymbol{H}_{i,j}. \tag{10}$$

The gradient of the negative log prior with respect to $\boldsymbol{H}$ is then

$$\nabla_H \mathcal{L}_{D,H}^{\text{NNSC}} = \lambda, \tag{11}$$

and the gradient with respect to $\boldsymbol{D}$ is zero, with the further normalization constraint given in Equation (9). □

2.3 Gaussian Process Priors

In the following, we derive the MAP estimate under the assumption that the non-negative matrices $\boldsymbol{D}$ and $\boldsymbol{H}$ are independently determined by a Gaussian process [18] connected by a link function. The Gaussian process framework provides a principled and practical approach to the specification of the prior probability distribution of the factors in the NMF model. The prior is specified in terms of two functions: i) a covariance function that describes corellations in the factors and ii) a link function, that transforms the Gaussian process prior into a desired distribution over the non-negative reals.

We assume that $\boldsymbol{D}$ and $\boldsymbol{H}$ are independent, so that we may write

$$\mathcal{L}_{D,H}(\boldsymbol{D}, \boldsymbol{H}) = \mathcal{L}_D(\boldsymbol{D}) + \mathcal{L}_H(\boldsymbol{H}). \tag{12}$$

In the following, we consider only the prior for $\boldsymbol{H}$, since the treatment of $\boldsymbol{D}$ is equivalent due to the symmetry of the NMF problem. We assume that there is an underlying variable vector, $\boldsymbol{h} \in \mathbb{R}^{LM}$, which is zero mean multivariate Gaussian with covariance matrix $\boldsymbol{\Sigma}_h$

$$\mathrm{p}_h(\boldsymbol{h}) = \left(2\pi|\boldsymbol{\Sigma}_h|^2\right)^{-\frac{1}{2}NL} \exp\left(-\frac{1}{2}\boldsymbol{h}^\top \boldsymbol{\Sigma}_h^{-1} \boldsymbol{h}\right), \tag{13}$$

and linked to $\boldsymbol{H}$ via a link function, f_h: $\mathbb{R}_+ \to \mathbb{R}$

$$\boldsymbol{h} = f_h\big(\mathrm{vec}\,(\boldsymbol{H})\big), \tag{14}$$

which operates element-wise on its input. The $\mathrm{vec}\,(\cdot)$ function in the expression stacks its matrix operand column by column. The link function should be strictly increasing, which ensures that the inverse exists. Later, we will further assume that the derivatives of f_h and f_h^{-1} exist. Under these assumptions, the prior over $\boldsymbol{H}$ is given by (using the change of variables theorem)

$$\begin{aligned} \mathrm{p}_H(\boldsymbol{H}) &= \mathrm{p}_h\Big(f_h\big(\mathrm{vec}\,(\boldsymbol{H})\big)\Big)\ \Big|\mathcal{J}\Big(f_h\big(\mathrm{vec}\,(\boldsymbol{H})\big)\Big)\Big| &(15)\\ &\propto \exp\left(-\frac{1}{2} f_h\big(\mathrm{vec}\,(\boldsymbol{H})\big)^\top \boldsymbol{\Sigma}_h^{-1} f_h\big(\mathrm{vec}\,(\boldsymbol{H})\big)\right) \prod_i \big|f_h'\big(\mathrm{vec}\,(\boldsymbol{H})\big)\big|_i, &(16)\end{aligned}$$

where $\mathcal{J}(\cdot)$ denotes the Jacobian determinant and f_h' is the derivative of the link function. The negative log prior is then

$$\mathcal{L}_H(\boldsymbol{H}) \propto \frac{1}{2} f_h\big(\mathrm{vec}\,(\boldsymbol{H})\big)^\top \boldsymbol{\Sigma}_h^{-1} f_h\big(\mathrm{vec}\,(\boldsymbol{H})\big) - \sum_i \log\big|f_h'\big(\mathrm{vec}\,(\boldsymbol{H})\big)\big|_i. \tag{17}$$

This expression can be combined with an appropriate likelihood function, such as the least squares likelihood in Equation (4), and be optimized to yield the MAP solution; however, in our experiments, we found that a more simple and robust algorithm can be obtained by making a change of variable as explained next.

2.4 Change of Optimization Variable

Instead of optimizing over the non-negative factors $\boldsymbol{D}$ and $\boldsymbol{H}$, we introduce the variables $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$, which are related to $\boldsymbol{D}$ and $\boldsymbol{H}$ by

$$\boldsymbol{D} = g_d(\boldsymbol{\delta}) = \mathrm{vec}^{-1}\left(f_d^{-1}(\boldsymbol{C}_d^\top \boldsymbol{\delta})\right), \quad \boldsymbol{H} = g_h(\boldsymbol{\eta}) = \mathrm{vec}^{-1}\left(f_h^{-1}(\boldsymbol{C}_h^\top \boldsymbol{\eta})\right), \tag{18}$$

where the $\mathrm{vec}^{-1}\,(\cdot)$ function maps its vector input into a matrix of appropriate size. The matrices $\boldsymbol{C}_d$ and $\boldsymbol{C}_h$ are the matrix square roots (Cholesky decompositions) of the covariance matrices $\boldsymbol{\Sigma}_d$ and $\boldsymbol{\Sigma}_h$, such that $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ are standard i.i.d. Gaussian.

This change of variable serves two purposes. First of all, we found that optimizing over the transformed variables was faster, more robust, and less prone to getting stuck in local minima. Second, the transformed variables are not constrained to be non-negative, which allows us to use existing unconstrained optimization methods to compute their MAP estimate.

The prior distribution of the transformed variable $\boldsymbol{\eta}$ is

$$\mathrm{p}_\eta(\boldsymbol{\eta}) = \mathrm{p}_H\big(g_h(\boldsymbol{\eta})\big)\ \big|\mathcal{J}\big(g_h(\boldsymbol{\eta})\big)\big| = \frac{1}{(2\pi)^{\frac{LM}{2}}} \exp\left(-\frac{1}{2}\boldsymbol{\eta}^\top \boldsymbol{\eta}\right), \tag{19}$$

and the negative log prior is

$$\mathcal{L}_\eta(\boldsymbol{\eta}) \propto \frac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\eta}. \tag{20}$$

To compute the MAP estimate of the transformed variables, we must combine this expression for the prior (and a similar expression for the prior of $\boldsymbol{\delta}$) with a likelihood function, in which the same change of variable is made

$$\mathcal{L}_{\delta,\eta|X}(\boldsymbol{\delta},\boldsymbol{\eta}) = \mathcal{L}_{X|D,H}\big(g_d(\boldsymbol{\delta}), g_h(\boldsymbol{\eta})\big) + \frac{1}{2}\boldsymbol{\delta}^\top\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\eta}^\top\boldsymbol{\eta}. \tag{21}$$

Then the MAP solution can be found by optimizing over $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$

$$\{\boldsymbol{\delta}_{\text{MAP}}, \boldsymbol{\eta}_{\text{MAP}}\} = \arg\min_{\boldsymbol{\delta},\boldsymbol{\eta}} \mathcal{L}_{\delta,\eta|X}(\boldsymbol{\delta},\boldsymbol{\eta}), \tag{22}$$

and, finally, estimates of $\boldsymbol{D}$ and $\boldsymbol{H}$ can be computed using Equation (18).

Example 3 (Least squares non-negative matrix factorization with Gaussian process priors (GPP-NMF))**.** *If we use the least squares likelihood in Equation (4), the posterior distribution in Equation (21) is given by*

$$\mathcal{L}^{\text{LS-GPP}}_{\delta,\eta|X}(\boldsymbol{\delta},\boldsymbol{\eta}) = \frac{1}{2}\left(\sigma_N^{-2}||\boldsymbol{X} - g_d(\boldsymbol{\delta})g_h(\boldsymbol{\eta})||_F^2 + \boldsymbol{\delta}^\top\boldsymbol{\delta} + \boldsymbol{\eta}^\top\boldsymbol{\eta}\right) \tag{23}$$

The MAP estimate of $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ is found by minimizing this expression, for which the derivative is useful

$$\nabla_\eta \mathcal{L}^{\text{LS-GPP}}_{\delta,\eta|X}(\boldsymbol{\delta},\boldsymbol{\eta}) = \sigma_N^{-2}\Big(\text{vec}\left(g_d(\boldsymbol{\delta})^\top(g_d(\boldsymbol{\delta})g_h(\boldsymbol{\eta}) - \boldsymbol{X})\right) \odot (f_h^{-1})'(\boldsymbol{C}_h^\top\boldsymbol{\eta})\Big)^\top \boldsymbol{C}_h + \boldsymbol{\eta}, \tag{24}$$

where $\odot$ denotes the Hadamard (element-wise) product. The derivative with respect to $\boldsymbol{\delta}$ is similar due to the symmetry of the NMF problem. □

2.5 Link Function

Any strictly increasing link function that maps the non-negative reals to the real line can be used in the proposed framework; however, in order to have a better probabilistic interpretation of the prior distribution, we propose a simple principle for choosing the link function. We choose the link function such that f_h^{-1} maps the marginal distribution of the elements of the underlying Gaussian process vector $\boldsymbol{h}$ into a specifically chosen marginal distribution of the elements of $\boldsymbol{H}$. Such an inverse function can be found as $f_h^{-1}(\boldsymbol{h}_i) = \text{P}_H^{-1}\big(\text{P}_h(\boldsymbol{h}_i)\big)$ where $\text{P}(\cdot)$ denotes the marginal cumulative distribution functions (cdf).

Since the marginals of a Gaussian process are Gaussian, $\text{P}_h(\boldsymbol{h}_i)$ is the Gaussian cdf, and, using Equation (13), the inverse link function is given by

$$f_h^{-1}(\boldsymbol{h}_i) = \text{P}_H^{-1}\left(\frac{1}{2} + \frac{1}{2}\Phi\left(\frac{\boldsymbol{h}_i}{\sqrt{2}\sigma_i}\right)\right) \tag{25}$$

where σ_i^2 is the i'th diagonal element of $\boldsymbol{\Sigma}_h$ and $\Phi(\cdot)$ is the error function.

Example 4 (Exponential-to-Gaussian link function)**.** *If we choose to have exponential marginals in* $\boldsymbol{H}$*, as in NNSC described in Example 2, we select* P_H *as the exponential cdf. The inverse link function is then*

$$f_h^{-1}(\boldsymbol{h}_i) = -\frac{1}{\lambda}\log\left(\frac{1}{2} - \frac{1}{2}\Phi\left(\frac{\boldsymbol{h}_i}{\sqrt{2}\sigma_i}\right)\right), \tag{26}$$

where λ *is an inverse scale parameter. The derivative of the inverse link function, which is needed for the parameter estimation, is given by*

$$(f_h^{-1})'(\boldsymbol{h}_i) = \frac{1}{\sqrt{2\pi}\sigma_i\lambda}\exp\left(\lambda f_h^{-1}(\boldsymbol{h}_i) - \frac{\boldsymbol{h}_i^2}{2\sigma_i^2}\right). \tag{27}$$

□

Example 5 (Rectified-Gaussian-to-Gaussian link function)**.** *Another interesting non-negative distribution is the rectified Gaussian given by*

$$p(x) = \begin{cases} \frac{2}{\sqrt{2\pi}s}\exp\left(-\frac{x^2}{2s^2}\right) & , \quad x \geq 0 \\ 0 & , \quad x < 0 \end{cases} \tag{28}$$

Using this pdf in Equation (25), the inverse link function is

$$f_h^{-1}(\boldsymbol{h}_i) = \sqrt{2}s\Phi^{-1}\left(\frac{1}{2} + \frac{1}{2}\Phi\left(\frac{\boldsymbol{h}_i}{\sqrt{2}\sigma_i}\right)\right), \tag{29}$$

and the derivative of the inverse link function is

$$(f_h^{-1})'(\boldsymbol{h}_i) = \frac{s}{2\sigma_i}\exp\left(\frac{f_h^{-1}(\boldsymbol{h}_i)^2}{2s^2} - \frac{\boldsymbol{h}_i^2}{2\sigma_i^2}\right). \tag{30}$$

□

2.6 Summary of the GPP-NMF Method

The GPP-NMF method can be summarized in the following steps.

1. Choose a suitable negative log likelihood function $\mathcal{L}_{X|D,H}(\boldsymbol{D}, \boldsymbol{H})$ based on knowledge of the distribution of the data or the residual.
2. For each of the non-negative factors $\boldsymbol{D}$ and $\boldsymbol{H}$, choose suitable link and covariance functions according to your prior beliefs. If necessary, draw samples from the prior distribution to examine its properties.
3. Compute the MAP estimate of $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ by Equation (22) using any suitable unconstrained optimization algorithm.
4. Compute $\boldsymbol{D}$ and $\boldsymbol{H}$ using Equation (18).

Our Matlab implemention of the GPP-NMF method is available online [23].

3 Experimental Results

We will demonstrate the proposed method on two examples, first a toy example, and second an example taken from the chemical shift brain imaging literature.

In our experiments we use the least squares GPP-NMF described in Example 3 and the link functions described in Example 4–5. The specific optimization method used to compute the GPP-NMF MAP estimate is not the topic of this paper, and any unconstrained optimization algorithm could in principle be used. In our experiments we used a simple gradient descent with line search to perform a total of 1000 alternating updates of $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$, after which the algorithm had converged. For details of the implementation, see the accompanying Matlab code [23].

3.1 Toy Example

We generated a 100×200 data matrix, $\boldsymbol{Y}$, by taking a random sample from the GPP-NMF model with two factors. We chose the generating covariance function for both $\boldsymbol{\delta}$ and $\boldsymbol{\eta}$ as a Gaussian radial basis function (RBF),

$$\phi(i,j) = \exp\left(-\frac{(i-j)^2}{\beta^2}\right), \tag{31}$$

where i and j are two sample indices, and the length scale parameter, which determines the smoothness of the factors, was $\beta^2 = 100$. We set the covariance between the two factors to zero, such that the factors were uncorrelated. For the matrix $\boldsymbol{D}$ we used the rectified-Gaussian-to-Gaussian link function with $s = 1$, and for $\boldsymbol{H}$ we used the exponential-to-Gaussian link function with $\lambda = 1$. Finally, we added independent Gaussian noise with variance $\sigma_N^2 = 25$, which resulted in a signal-to-noise ratio of approximately -7 dB. The generated data matrix is shown in Figure 1.

We then decomposed the generated data matrix using four different methods:

1. **LS-NMF:** Standard least squares NMF [8]. This algorithm does not allow negative data points, so these were set to zero in the experiment.

2. **CNMF:** Constrained NMF [6, 7], which is a least squares NMF algorithm that allows negative observations.

3. **GPP-NMF: Correct prior:** The proposed method with correct link-functions, covariance matrix, and parameter values.

4. **GPP-NMF: Incorrect prior:** To illustrate the sensitivity of the method to prior assumptions, we evaluated the proposed method with incorrect prior assumptions: We switched the link functions, such that for $\boldsymbol{D}$ we used the exponential-to-Gaussian, and for $\boldsymbol{H}$ we used the rectified-Gaussian-to-Gaussian. We used an RBF covariance function with $\beta^2 = 10$ and $\beta^2 = 1000$ for $\boldsymbol{D}$ and $\boldsymbol{H}$ respectively.

The results of the decompositions are shown as reconstructed data matrices in Figure 1. All four methods find solutions that visually appear to fit the underlying data. Both LS-NMF and CNMF find non-smooth solutions, whereas the two GPP-NMF results are smooth in accordance with the priors. In the GPP-NMF with incorrect prior, the dark areas (high pixel intensities) appear too wide in the first axis direction and too narrow in the section axis direction, due to the incorrect setting of the covariance function. The GPP-NMF with correct prior is visually almost equal to the true underlying data.

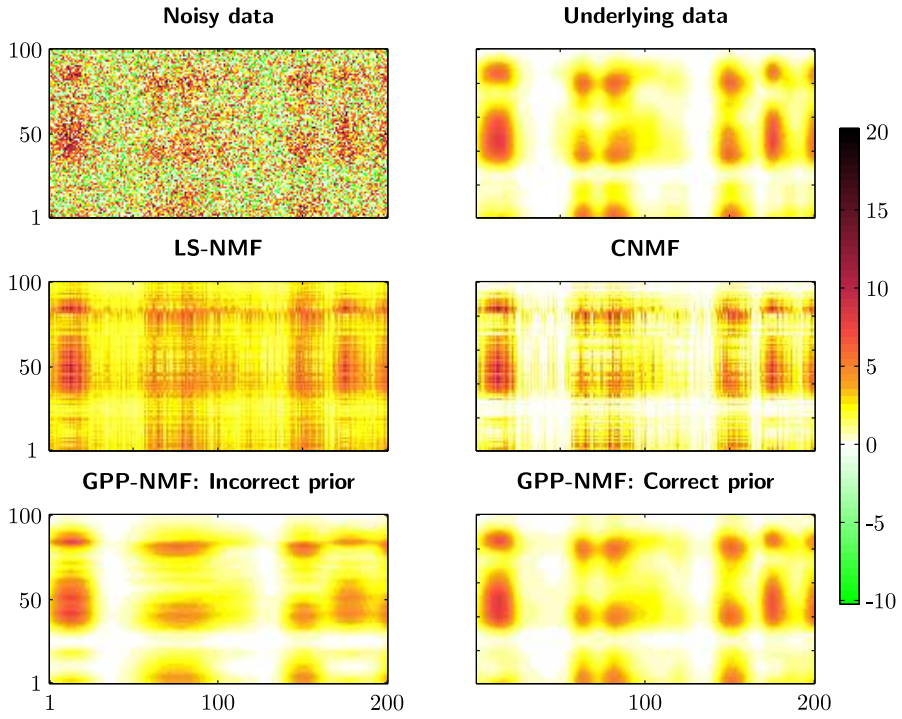

Figure 1: Toy example data matrix (upper left), underlying noise-free non-negative data (upper right), and estimates using the four methods described in the text. The data has a fairly large amount of noise and the underlying non-negative factors are smooth in both directions. The LS-NMF and CNMF decomposition are non-smooth, since these methods do not model of correlations in the factors. The GPP-NMF, which uses a smooth prior, finds a smooth solution. When using the correct prior, the soulution is very close to the true underlying data.

Plots of the estimated factors are show in Figure 2. The factors estimated by the LS-NMF and the CNMF methods appear noisy and are non-smooth, whereas the factors estimated by the GPP-NMF are smooth. The factors estimated by the LS-NMF method have a positive bias, because of the truncation of negative

data. The GPP-NMF with incorrect prior has too many local extrema in the $\boldsymbol{D}$ factor and too few in the $\boldsymbol{H}$ factor due to the incorrect covariance functions. There are only minor difference between the result of the GPP-NMF with the correct prior and the underlying factors.

Measures of root mean squared error (RMSE) of the four decompositions are given in Figure 3. All four methods fit the noisy data almost equally well. (Note that, due to the additive noise with variance 25, a perfect fit to the underlying factors would result in a RMSE of 5 with respect to the noisy data.) The LS-NMF fits the data worst due to the truncation of negative data points, and the CNMF fits the data best, due to overfitting. With respect to the noise free data and the underlying factors, the RMSE is worst for the LS-NMF and best for the GPP-NMF with correct prior. The GPP-NMF with incorrect prior is better than both LS-NMF and CNMF in this case. This shows, that in this situation it better to use a prior which is not perfectly correct, compared to using no prior as in the LS-NMF and CNMF methods, (which corresponds to a flat prior over the non-negative reals and no correlations.)

3.2 Chemical Shift Brain Imaging Example

Next, we demonstrate the GPP-NMF method on ^{1}H decoupled ^{31}P chemical shift imaging data of the human brain. We use the data set from Ochs et al. [24], which has also been analyzed by Sajda et al. [6, 7]. The data set, which is shown in Figure 4, consists of 512 spectra measured on an $8 \times 8 \times 8$ grid in the brain.

Ochs et al. [24] use PCA to determine, that the data set is adequately described by two sources (which correspond to brain and muscle tissue.) They propose a bilinear Bayesian approach, in which they use a smooth prior over the constituent spectra, and force to zero the amplitude of the spectral shape corresponding to muscle tissue at 12 positions deep inside the head. Their approach produces physically plausible results, but it is computationally very expensive and takes several hours to compute.

Sajda et al. [6, 7] propose an NMF approach that is reported also to produce physically plausible results. Their method is several orders of magnitude faster, taking less than a second to compute. The disadvantage of the method of Sajda et al. compared to the Bayesian approach of Ochs et al. is, that it provides no mechanism for using prior knowledge to improve the solution.

The GPP-NMF approach we propose in this paper bridges the gap between the two previous approaches, in the sense that it is a relatively fast NMF approach, in which priors over the factors can be specified. These priors are specified by the choice of the link and covariance functions. We used prior predictive sampling to find reasonable settings of the the function parameters: We drew random samples from the prior distribution and examined properties of the factors and reconstructed data. We then manually adjusted the parameters of the prior to match our prior beliefs. An example of a random draw from the prior distribution is shown in Figure 5, with the parameters set as described below.

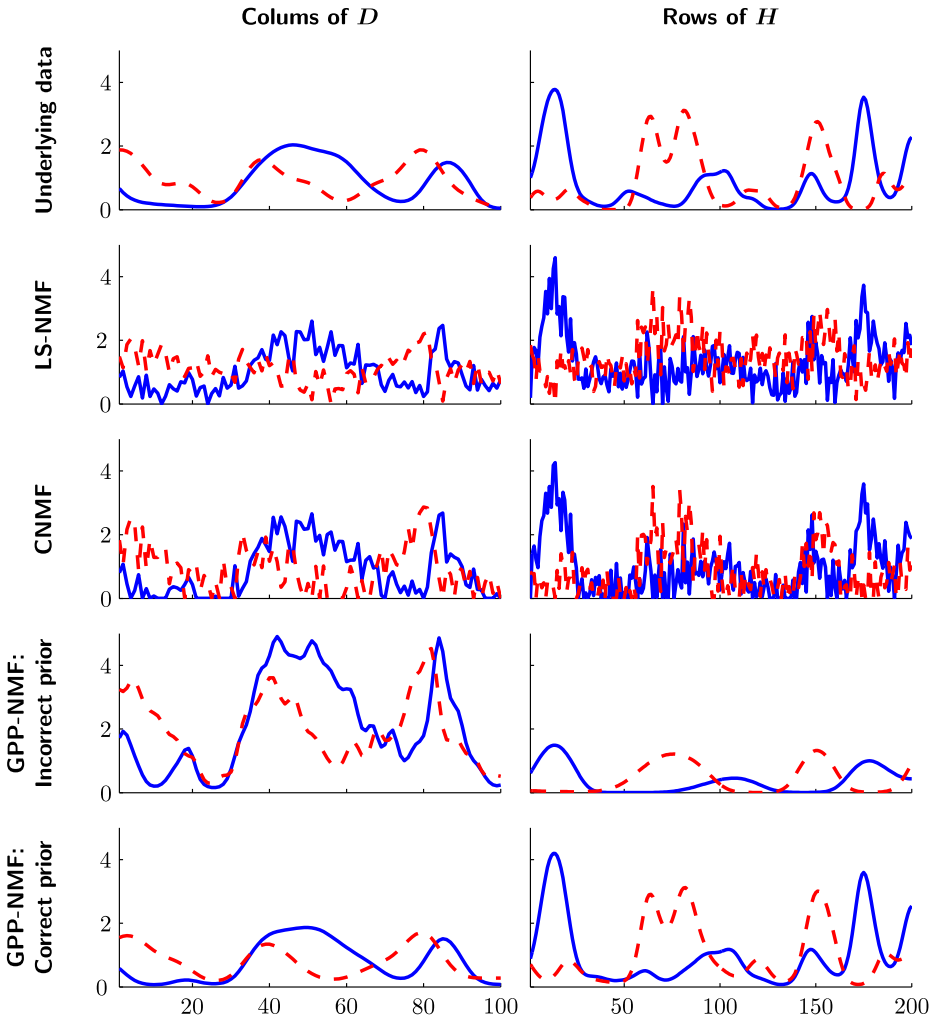

Figure 2: Underlying non-negative factors in the toy example: Columns of $\boldsymbol{D}$ (left) and rows of $\boldsymbol{H}$ (right). The factors found by the LS-NMF and the CNMF algorithm are noisy, whereas the factors found by the GPP-NMF method are smooth. When using the correct prior, the factors found are very similar to the true factors.

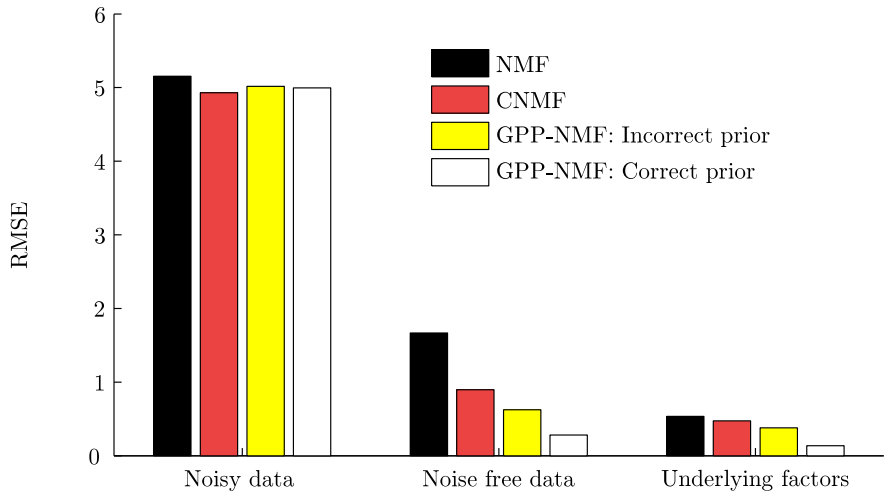

Figure 3: Toy example: Root mean squared error (RMSE) with respect to the noisy data, the underlying noise free data, and the true underlying non-negative factors. The CNMF solution fits the noisy data slightly better, but the GPP-NMF solution fits the underlying data much better.

We assumed that the factors are uncorrelated, so the covariance between factors is zero. We used a Gaussian RBF covariance function for the constituent spectra, with a length scale of $\beta = 0.3$ parts per million (ppm), and we used the exponential-to-Gaussian link function with $\lambda_d = 1$. This gave a prior for the spectra that is sparse with narrow smooth peaks. For the amplitude at the 512 voxels in the head, we used a Gaussian RBF covariance function on the 3-D voxel indices, with length scale $\beta = 2$. Furthermore, we centered the left-to-right coordinate axis in the middle of the brain, and computed the RBF kernel on the absolute value of this coordinate, so that a left-to-right symmetry was introduced in the prior distribution. Again, we used the exponential-to-Gaussian link function, and we chose $\lambda_h = 2 \cdot 10^{-4}$ to match the overall magnitude of the data. This gave a prior for the amplitude distribution that is sparse, smooth, and symmetric. The noise variance was set to $\sigma_N^2 = 10^8$ which corresponds to the noise level in the data set.

We then decomposed the data set using the proposed GPP-NMF algorithm and, for comparison, reproduced the results of Sajda et al. [7] using their CNMF method. The results of the experiments are shown in Figure 4. An example of a random draw from the prior distribution is shown in Figure 5. The results of the CNMF is shown in Figure 6, and the results of the GPP-NMF is shown in Figure 7. The figures show the constituent spectra and the fifth axial slice of the spatial distribution of the spectra. The 8×8 spatial distributions are smoothed in the illustration, similar to the way the results are visualized in the literature.

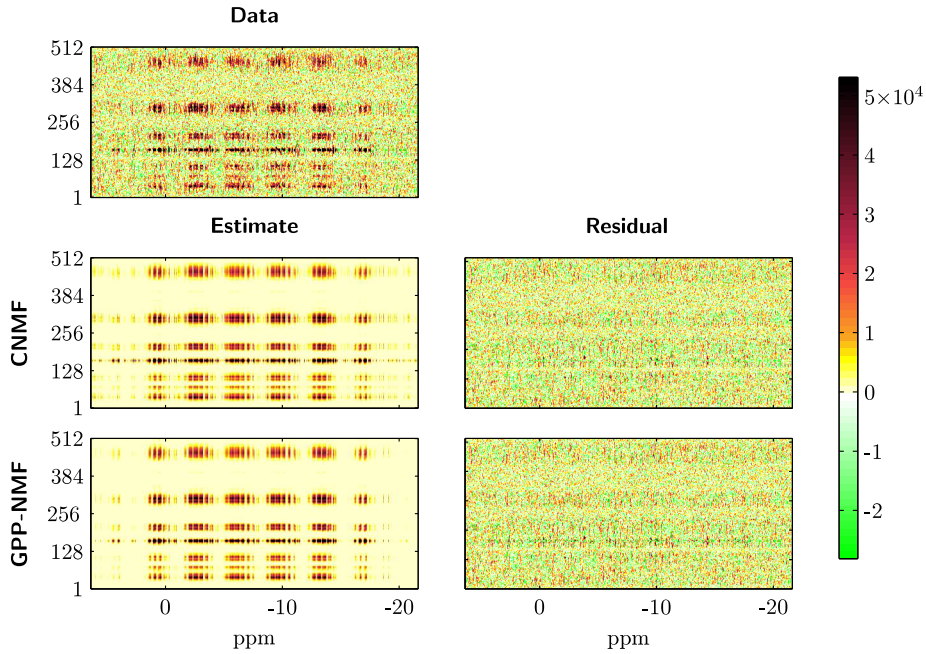

Figure 4: Brain imaging data matrix (top) along with the estimated decomposition and residual for the CNMF (middle) and GPP-NMF (bottom) method. In this view the results of the two decompositions are very similar, the data appears to be modeled equally well and the residuals are similar in magnitude.

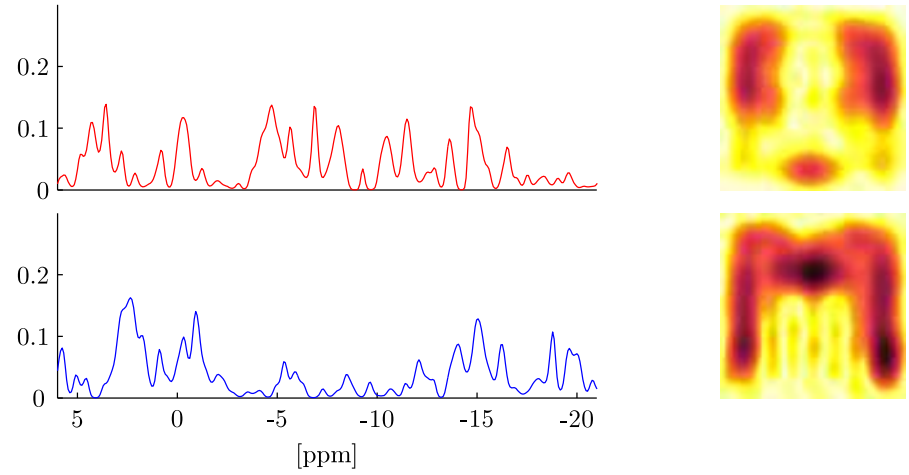

Figure 5: Brain imaging data: Random draw from the prior distribution with the parameters set as described in the text. The prior distribution of the constituent spectra (left) is exponential and smooth and the spatial distribution (right) in the brain is exponential, smooth, and has a left-to-right symmetry.

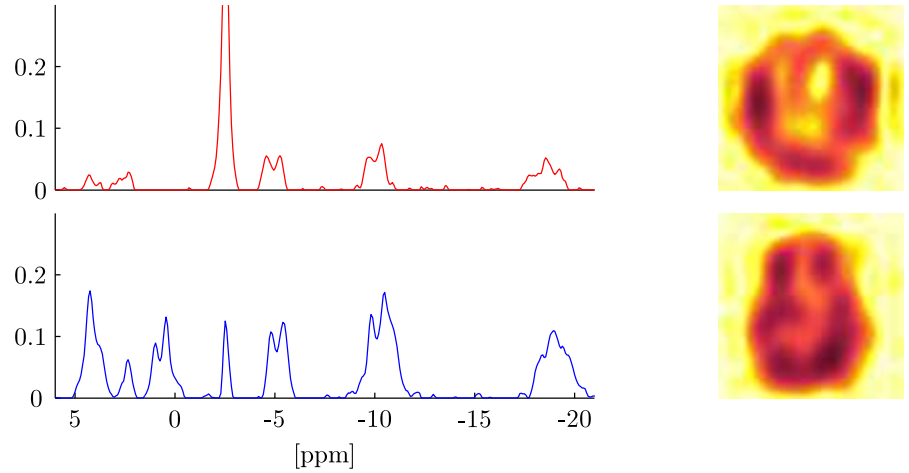

Figure 6: CNMF decomposition result. The recovered spectra are physically plausible, and the spatial distribution in the brain for the muscle (top) and brain (bottom) tissue is somewhat separated. Muscle tissue is primarily found near the edge of the skull, whereas brain tissue is primarliy found at the inside of the head.

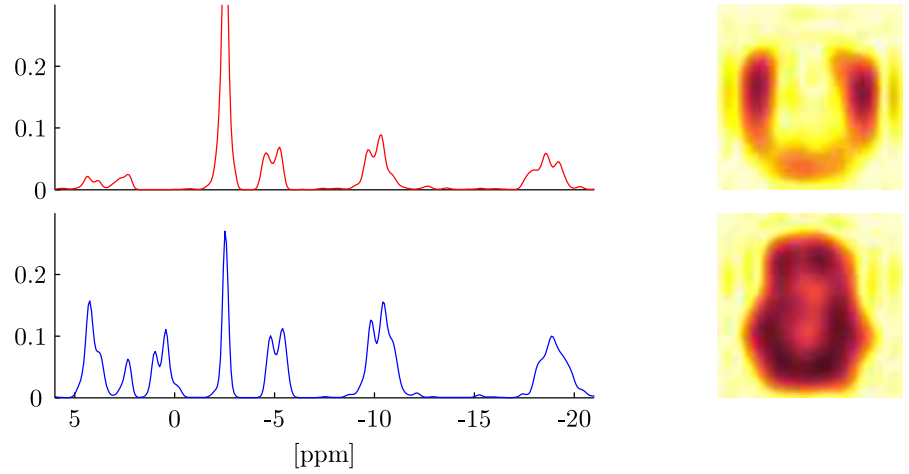

Figure 7: GPP-NMF decomposition result. The recovered spectra are very similar to the spectra found by the CNMF method, but they are slightly more smooth. The spatial distribution in the brain is highly separated between brain and muscle tissue, and it is more symmetric than the CNMF solution.

The results show that both methods give physically plausible results. The main difference is that the spatial distribution of the spectra corresponding to muscle and brain tissue is much more separated in the GPP-NMF result, which is due to the exponential, smooth, and symmetric prior distribution. By including prior information, we obtain a solution, where the factor corresponding to muscle tissure is clearly located on the edge of the skull.

4 Conclusions

We have introduced a general method for exploiting prior knowledge in non-negative matrix factorization, based on Gaussian process priors, linked to the non-negative factors by a link function. The method can be combined with any existing NMF cost function that has a probabilistic interpretation, and any existing unconstrained optimization algorithm can be used to compute the maximum a posteriori estimate.

Experiments on toy data show, that with a suitable selection of the prior distribution of the non-negative factors, the GPP-NMF method gives much better results in terms of estimating the true underlying factors, both when compared to traditional NMF and CNMF.

Experiments on chemical shift brain imaging data show that the GPP-NMF method can be successfully used to include prior knowledge of the spectral and spatial distribution, resulting in better spatial separation between spectra corresponding to muscle and brain tissue.

5 Acknowledgments

We would like to thank Paul Sajda and Truman Brown for making the brain imaging data available to us. This research was supported by the Intelligent Sound project, Danish Technical Research Council grant no. 26–02–0092 and partly also by the European Commission through the sixth framework IST Network of Excellence: Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL), contract no. 506778.

References

[1] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error-estimates of data values," *Environmetrics*, vol. 5, pp. 111–126, 1994.

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

[3] L. Weixiang, Z. Nanning, and Y. Qubo, "Nonnegative matrix factorization and its applications in pattern recognition," *Chinese Science Bulletin*, vol. 51, no. 1, pp. 7–18, Jan 2006.

[4] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Data Mining, Proceedings of SIAM International Conference on*, 2005.

[5] S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita, "Dimensionality reduction using non-negative matrix factorization for information retrieval," in *Systems, Man, and Cybernetics, IEEE International Conference on*, vol. 2, 2001, pp. 960–965.

[6] P. Sajda, S. Du, and L. Parra, "Recovery of constituent spectra using non-negative matrix factorization." in *Wavelets: Applications in Signal and Image Processing, Proceedings of SPIE*, vol. 5207, 2003, pp. 321–331.

[7] P. Sajda, S. Du, T. R. Brown, R. Stoyanova, D. C. Shungu, X. Mao, and L. C. Parra, "Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain," *Transactions on Medical Imaging, IEEE*, vol. 23, no. 12, pp. 1453–1465, Dec 2004.

[8] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Neural Information Processing Systems, Advances in (NIPS)*, 2000, pp. 556–562.

[9] P. Hoyer, "Non-negative sparse coding," in *Neural Networks for Signal Processing, IEEE Workshop on*, 2002, pp. 557–565.

[10] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Computer Vision and Pattern Recognition, Proceedings of the IEEE Computer Society Conference on*, vol. 1, Dec 2001, pp. 207–212.

[11] Z. Chen and A. Cichocki, "Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints," Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep., 2005.

[12] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *International Computer Music Conference, ICMC*, 2003.

[13] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sourses from monophonic inputs," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA), Lecture Notes in Computer Science*, vol. 3195, Sep 2004, pp. 494–499.

[14] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-d deconvolution for blind single channel source separation," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA), Lecture Notes in Computer Science*, Apr 2006, vol. 3889, pp. 700–707.

[15] O. Winther and K. B. Petersen, "Bayesian independent component analysis: Variational methods and non-negative decompositions," *Digital Signal Processing*, vol. 17, no. 5, 2007.

[16] T. Hofmann, "Probabilistic latent semantic indexing," in *Research and Development in Information Retrieval, Proceedings of the International SIGIR Conference on*, 1999.

[17] A. Cichocki, R. Zdunek, and S. ichi Amari, "Csiszar's divergences for non-negative matrix factorization: Family of new algorithms," in *Lecture Notes in Computer Science*, vol. 3889, 2006, pp. 32–39.

[18] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[19] C.-J. Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, pp. 2756–2779, 2007.

[20] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization."

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

[22] J. Eggert and E. Körner, "Sparse coding and NMF," in *Neural Networks, IEEE International Conference on*, vol. 4, 2004, pp. 2529–2533.

[23] M. N. Schmidt. (2008) Non-negative matrix factorization with gaussian process priors. [Online]. Available: http://www.mikkelschmidt.dk/cin2008

[24] M. F. Ochs, R. S. Stoyanova, F. Arias-Mendoza, and T. R. Brown, "A new method for spectral decomposition using a bilinear bayesian approach," *Journal of Magnetic Resonance*, pp. 161–176, 1999.

Bibliography

[1] S. A. Abdallah and M. D. Plumbley, "Polyphonic transcription by non-negative sparse coding of power spectra," in *Music Information Retrieval, International Conference on (ISMIR)*, Oct 2004, pp. 318–325.

[2] T. Abe, T. Kobayashi, and S. Imai, "Harmonics tracking and pitch extraction based on instantaneous frequency," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 1, May 1995, pp. 756–759.

[3] J.-H. Ahn, S.-K. Kim, J.-H. Oh, and S. Choi, "Multiple nonnegative-matrix factorization of dynamic pet images," in *Computer Vision, Asian Conference on*, Jan 2004, pp. 1009–1013.

[4] R. Albright, J. Cox, D. Duling, A. Langville, and C. Meyer, "Algorithms, initializations, and convergence for the nonnegative matrix factorization," North Carolina State University, Tech. Rep. 81706, 2006.

[5] C. Andrieu and A. Doucet, "Stochastic algorithms for marginal MAP retrieval of sinusoids in non-Gaussian noise," in *Statistical Signal and Array Processing, IEEE Workshop on*, 2000, pp. 131–135.

[6] P. Anttila, P. Paatero, U. Tapper, and O. Järvinen, "Source identification of bulk wet deposition in Finland by positive matrix factorization," *Atmospheric Environment*, vol. 29, no. 14, pp. 1705–1718, 1995.

[7] H. Asari, "Auditory system characterization," Ph.D. dissertation, Watson School of Biological Sciences, Jul 2007.

[8] H. Asari, "Non-negative matrix factorization: A possible way to learn sound dictionaries," Tony Zador Lab, Watson School of Biological Sciences, Cold Spring Harbor Laboratory, Tech. Rep., Aug 2005.

[9] F. R. Bach and M. I. Jordan, "Learning spectral clustering, with application to speech separation," *Journal of Machine Learning Research*, vol. 7, pp. 1963–2001, 2006.

[10] F. R. Bach and M. I. Jordan, "Learning spectral clustering," in *Neural Information Processing Systems, Advances in (NIPS)*, 2004.

[11] F. R. Bach and M. I. Jordan, "Blind one-microphone speech separation: A spectral learning approach," in *Neural Information Processing Systems, Advances in (NIPS)*, 2005, pp. 65–72.

[12] L. Badea, "Clustering and metaclustering with nonnegative matrix decompositions," in *Machine Learning, European Conference on (ECML), Lecture Notes in Computer Science (LNCS)*, vol. 3720. Springer, Nov 2005, pp. 10–22.

[13] R. Balan and J. Rosca, "A spectral power factorization," Siemens Corporate Research. Princeton, NJ, Tech. Rep. SCR-01-TR-703, Sep 2001.

[14] R. Balan, A. Jourjine, and J. Rosca, "Ar process and sources can be reconstructed from degenerate mixtures," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, Jan 1999, pp. 467–472.

[15] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Gosh, "Clustering with bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, Oct 2005.

[16] S. Behnke, "Discovering hierarchical speech features using convolutional non-negative matrix factorization," in *Neural Networks (IJCNN), Proceeding of the International Joint Conference on*, vol. 4, Jul 2003, pp. 2758–2763.

[17] T. Beierholm, B. D. Pedersen, and O. Winther, "Low complexity bayesian single channel source separation," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 5, May 2004, pp. 529–532.

[18] L. Benaroya and F. Bimbot, "Wiener based source separation with hmm/gmm using a single sensor," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, Apr 2003.

[19] L. Benaroya, L. M. Donagh, F. Bimbot, and R. Gribonval, "Non negative sparse representation for wiener based source separation with a single sensor," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 6, Apr 2003, pp. 613–616.

[20] L. Benaroya, F. Bimbot, and R. Gribonval, "Audio source separation with a single sensor," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 191–199, Jan 2006.

[21] M. W. Berry and M. Brown, "Email surveillance using nonnegative matrix factorization," *Computational and Mathematical Organization Theory*, vol. 11, pp. 249–264, Feb 2005.

[22] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics and Data Analysis*, 2006.

[23] D. P. Bertsekas, "Projected newton methods for optimization problems with simple constraints," *Control and Optimization, Siam Journal on*, vol. 20, no. 2, pp. 221–246, Mar 1982.

[24] J. C. Bezdek and R. J. Hathaway, "Some notes on alternating optimization," in *Fuzzy Systems, AFSS International Conference on*, ser. Lecture Notes in Computer Science, vol. 2275. Springer, Jan 2002, pp. 187–195.

[25] J. C. Bezdek, R. J. Hathaway, R. E. Howard, C. A. Wilson, and M. P. Windham, "Local convergence analysis of a grouped variable version of coordinate descent," *Optimization Theory and Applications, Journal of*, vol. 54, no. 3, pp. 471–477, Sep 1987.

[26] R. Blouet, G. Rapaport, I. Cohen, and C. Févotte, "Evaluation of several strategies for single sensor speech/music separation," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, Apr 2008.

[27] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal—part 1: Fundamentals," *Proceedings of the IEEE*, vol. 80, no. 4, pp. 520–538, Apr 1992.

[28] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization," *Pattern Recognition*, vol. 41, no. 4, pp. 1350–1362, 2008.

[29] R. Bro and S. D. Jong, "A fast non-negativity-constrained least squares algorithm," *Chemometrics, Journal of*, vol. 11, no. 5, pp. 393–401, 1999.

[30] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences (PNAS)*, vol. 101, no. 12, pp. 4164–4169, Mar 2004.

[31] G. Buchsbaum and O. Bloch, "Color categories revealed by non-negative matrix factorization of munsell color spectra," *Vision Research*, vol. 42, no. 5, pp. 559–563, Mar 2002.

[32] I. Buciu and I. Pitas, "Application of non-negative and local non negative matrix factorization to facial expression recognition," in *Pattern Recognition, International Conference on (ICPR)*, vol. 1, Aug 2004, pp. 288–291.

[33] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *Scientific and Statistical Computing, SIAM Journal on*, vol. 16, no. 5, pp. 1190–1208, 1995.

[34] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "L-bfgs-b, fortran subroutines for large-scale bound constrained optimization," *Mathematical Software, ACM Transactions on*, vol. 23, no. 4, pp. 550–560, 1997.

[35] J. D. Carroll and J. J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of eckart-young decomposition," *Psychometrika*, vol. 35, pp. 283–319, 1970.

[36] M. Catral, L. Han, M. Neumann, and R. J. Plemmons, "On reduced rank nonnegative matrix factorization for symmetric nonnegative matrices," *Linear Algebra and its Applications*, vol. 393, pp. 107–126, Dec 2004.

[37] Z. Chen and A. Cichocki, "Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints," Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep., 2005.

[38] Z. Chen, A. Cichocki, and T. M. Rutkowski, "Constrained non-negative matrix factorization method for eeg analysis in early detection of alzheimer disease," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, May 2006.

[39] Y.-C. Cho, S. Choi, and S.-Y. Bang, "Non-negative component parts of sound for classification," in *Signal Processing and Information Technology, IEEE International Symposium on (ISSPIT)*, Dec 2003, pp. 633–636.

[40] Y.-C. Cho and S. Choi, "Learning nonnegative features of spectrotemporal sounds for classification," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1327–1336, Jul 2005.

[41] M. T. Chu and M. M. Lin, "Low dimensional polytope approximation and its applications to nonnegative matrix factorization," *SIAM Journal on Scientific Computing*, pp. 1131–1155, Mar 2008.

[42] M. Chu, F. Diele, R. J. Plemmons, and S. Ragni, "Optimality, computation, and interpretations of nonnegative matrix factorizations," Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, Tech. Rep., 2004.

[43] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization," *Electronic Letters*, vol. 42, no. 16, pp. 947–958, 2006.

[44] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorization using projected gradient approaches," in *Neural Information Processing, International Conference on (ICONIP)*, Oct 2006.

[45] A. Cichocki and R. Zdunek, "Regularized alternating least squares algorithms for non-negative matrix/tensor factorization," in *Neural Networks, Advances in (ISSN), Lecture Notes in Computer Science*, vol. 4493, 2007, pp. 793–802.

[46] A. Cichocki and R. Zdunek, "Nonnegative matrix and tensor factorization," *Signal Processing Magazine, IEEE*, vol. 25, no. 1, pp. 142–145, Jan 2008.

[47] A. Cichocki, S. ichi Amari, R. Zdunek, R. Kompass, G. Hori, and Z. He, "Extended smart algorithms for non-negative matrix factorization," in *Artificial Intelligence and Soft Computing, International Conference on (ICAISC)*, vol. 4029, Jun 2006, pp. 548–562.

[48] A. Cichocki, R. Zdunek, and S. ichi Amari, "New algorithms for non-negative matrix factorization in applications to blind source separation," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 5, May 2006, pp. 621–625.

[49] A. Cichocki, R. Zdunek, and S. ichi Amari, "Csiszár's divergences for non-negative matrix factorization: Family of new algorithms," in *Lecture Notes in Computer Science (LNCS)*, vol. 3889. Springer, 2006, pp. 32–39.

[50] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S. ichi Amari, "Novel multi-layer non-negative tensor factorization with sparsity constraints," in *Adaptive and Natural Computing Algorithms (ICANNGA)*, vol. 4432, Apr 2007, pp. 271–280.

[51] A. Cichocki, R. Zdunek, R. plemmons, and S. ichi Amari, "Non-negative tensor factorization using alpha and beta divergences," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, Apr 2007, pp. 1393–1396.

[52] M. Cooper and J. Foote, "Summarizing video using non-negative similarity matrix factorization," in *Multimedia Signal Processing, IEEE Workshop on*, Dec 2002, pp. 22–28.

[53] A. Cutler and L. Breiman, "Archetypal analysis," *Technometrics*, vol. 36, no. 4, pp. 338–347, Nov 1994.

[54] I. S. Dhillon and S. Sra, "Generalized nonnegative matrix approximations with bregman divergences," University of Texas at Austin, Department of Computer Sciences, Tech. Rep., 2005.

[55] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Data Mining, Proceedings of SIAM International Conference on*, 2005, pp. 606–610.

[56] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorization," Lawrence Berkeley National Laboratory, University of California, Berkeley, Tech. Rep. 60428, Nov 2006.

[57] C. Ding, T. Li, and W. Peng, "Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence, chi-square statistic, and a hybrid method," in *Artificial Intelligence, AAAI National Conference on*, Jul 2006.

[58] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Knowledge Discovery and Data Mining, International Conference on*, 2006, pp. 126–135.

[59] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Neural Information Processing Systems, Advances in (NIPS)*, 2003.

[60] A. Doucet, S. J. Godsill, and C. P. Robert, "Marginal maximum a posteriori estimation using markov chain monte carlo," *Statistics and Computing*, vol. 12, no. 1, pp. 77–84, Jan 2002.

[61] J. Eggert and E. Körner, "Sparse coding and NMF," in *Neural Networks, IEEE International Conference on*, vol. 4, 2004, pp. 2529–2533.

[62] J. Eggert, H. Wersing, and E. Körner, "Transformation-invariant representation and nmf," in *Neural Networks (IJCNN), Proceeding of the International Joint Conference on*, 2004, pp. 2535–2539.

[63] D. P. W. Ellis and D. Rosenthal, "Mid-level representations for computational auditory scene analysis," in *Artificial Intelligence, International Joint Conference on (IJCAI)*, Aug 1995.

[64] D. P. W. Ellis and R. J. Weiss, "Model-based monaural source separation using a vector-quantized phase-vocoder representation," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, May 2006, pp. 957–960.

[65] T. Feng, S. Z. Li, H.-Y. Shum, and H. Zhang, "Local non-negative matrix factorization as a visual representation," in *Development and Learning, International Conference on*, 2002, pp. 178–183.

[66] D. FitzGerald, M. Cranitch, and E. Coyle, "Shifted 2d non-negative tensor factorization," in *Statistics in Signal Processing, IEEE Conference on*, Jul 2005.

[67] D. FitzGerald, M. Cranitch, and E. Coyle, "Extended nonnegative tensor factorisation models for musical sound source separation," *Computational Intelligence and Neuroscience*, Apr 2008.

[68] I. K. Fodor, "A survey of dimension reduction techniques," Lawrence Livermore National Laboratory, Tech. Rep., 2002.

[69] P. Fogel, S. S. Young, D. M. Hawkins, and N. Ledirac, "Inferential, robust non-negative matrix factorization analysis of microarray data," *Bioinformatics*, vol. 23, no. 1, pp. 44–49, 2007.

[70] T. Fujiwara, S. Ishikawa, Y. Hoshida, K. Inamura, T. Isagawa, M. Shimane, H. Aburatani, Y. Ishikawa, and H. Nomura, "Non-negative matrix factorization of lung adenocarcinoma expression profiles," in *Genome Informatics, International Conference on*, Dec 2005.

[71] S. A. Fulop and K. Fitz, "Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications," *Journal of the Acoustical Society of America*, vol. 119, pp. 360–371, Jan 2006.

[72] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization," *Bioinformatics*, vol. 21, pp. 3970–3975, 2005.

[73] A. Garrido Frenich, M. Martnez Galera, J. Martnez Vidal, D. Massart, J. Torres-Lapasio, K. De Braekeleer, J.-H. Wang, and P. Hopke, "Resolution of multicomponent peaks by orthogonal projection approach, positive matrix factorization and alternating least squares," *Analytica Chimica Acta*, vol. 411, no. 1–2, pp. 145–155, 2000.

[74] E. Gaussier and C. Goutte, "Relation between PLSA and NMF and implication," in *Research and Development in Information Retrieval, Proceedings of the International SIGIR Conference on*, 2005, pp. 601–602.

[75] Z. Ghahramani and M. I. Jordan, "Factorial hidden markov models," *Machine Learning*, vol. 29, pp. 245–273, 1997.

[76] O. Gillet and G. Richard, "Transcription and separation of drum signals from polyphonic music," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 3, pp. 529–540, Mar 2008.

[77] E. F. Gonzalez and Y. Zhang, "Accelerating the lee-seung algorithm for nonnegative matrix factorization," Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005, Tech. Rep. TR05-02, 2005.

[78] D. Guillamet, B. Schiele, and J. Vitria, "Analyzing non-negative matrix factorization for image classification," in *Pattern Recognition, International Conference on (ICPR)*, vol. 2, Aug 2002, pp. 116–119.

[79] D. Guillamet and J. Vitrià, "Non-negative matrix factorization for face recognition," in *Topics in Artificial Intelligence*, ser. Lecture Notes in Computer Science (LNCS). Springer, 2002, vol. 2504, pp. 336–344.

[80] D. Guillamet and J. Vitrià, "Classifying faces with non-negative matrix factorization," in *Artificial Intelligence, Catalonian Conference on (CCIA)*, 2002, pp. 24–31.

[81] D. Guillamet, M. Bressan, and J. Vitrià, "A weighted non-negative matrix factorization for local representations," in *Computer Vision and Pattern Recognition, IEEE Conference on (CVPR)*, vol. 1, 2001, pp. 942–947.

[82] D. Guillamet, J. Vitrià, and B. Schiele, "Introducing a weighted non-negative matrix factorization for image classification," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2447–2454, Oct 2003.

[83] A. Hamza and D. Brady, "Reconstruction of reflectance spectra using robust nonnegative matrix factorization," *Signal Processing, IEEE Transactions on*, vol. 54, no. 9, pp. 3637–3642, 2006.

[84] L. K. Hansen and K. B. Petersen, "Monaural ica of white noise mixtures is hard," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, Apr 2003, pp. 815–820.

[85] R. A. Harshman, "Foundations of the parafac procedure: Models and conditions for an "explanatory" multi modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[86] R. A. Harshman, "Parafac2: Mathematical and technical notes," *UCLA Working Papers in Phonetics*, vol. 22, pp. 30–47, 1972.

[87] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3d non-negative tensor factorization," in *Computer Vision, IEEE International Conference on*, vol. 1, Oct 2005, pp. 50–57.

[88] M. Heiler and C. Schnörr, "Controlling sparseness in nonnegative tensor factorization," in *Computer Vision, European Conference on (ECCV)*, ser. Lecture Notes in Computer Science (LNCS), vol. 3951. Springer, 2006, pp. 56–67.

[89] M. Heiler and C. Schnörr, "Learning sparse representations by non-negative matrix factorization and sequential cone programming," *Journal of Machine Learning Research*, vol. 7, pp. 1385–1407, Jul 2006.

[90] M. Helén and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," in *European Signal Processing Conference, Proceedings of (EUSIPCO)*, Sep 2005.

[91] F. L. Hitchcock, "Multiple invariants and generalized rank of a p-way matrix or tensor," *Mathematics and Physics, Journal of*, vol. 7, no. 1, pp. 39–70, 1927.

[92] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Mathematics and Physics, Journal of*, vol. 6, no. 1, pp. 164–189, 1927.

[93] T. Hofmann, "Probabilistic latent semantic indexing," in *Research and Development in Information Retrieval, Proceedings of the International SIGIR Conference on*, 1999.

[94] T. Hofmann, "Probabilistic latent semantic analysis," in *Uncertainty in Artificial Intelligence (UAI)*, 1999, pp. 289–296.

[95] J. R. Hopgood and P. J. W. Rayner, "Single channel nonstationary stochastic signal separation using linear time-varying filters," *Signal Processing, IEEE Transactions on*, vol. 51, no. 7, pp. 1739–1752, Jul 2003.

[96] P. O. Hoyer, "Non-negative sparse coding," in *Neural Networks for Signal Processing, IEEE Workshop on*, 2002, pp. 557–565.

[97] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, Nov 2004.

[98] P. O. Hoyer, "Modeling receptive fields with non-negative sparse coding," *Neurocomputing*, vol. 52–54, pp. 547–552, Jun 2003.

[99] C. Hu, B. Zhang, S. Yan, Q. Yang, J. Yan, Z. Chen, and W.-Y. Ma, "Mining ratio rules via principal sparse non-negative matrix factorization," in *Data Mining, IEEE/WIC/ACM International Conference on (ICDM)*, Nov 2004, pp. 407–410.

[100] G. Hu and D. Wang, "On amplitude modulation for monaural speech segregation," in *Neural Networks (IJCNN), Proceeding of the International Joint Conference on*, 2002, pp. 69–74.

[101] G. Hu and D. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation," *Neural Networks, IEEE Transactions on*, vol. 15, no. 5, pp. 1135–1150, Sep 2004.

[102] G.-J. Jang and T.-W. Lee, "Monaural source separation," in *Blind Speech Separation*, S. Makino, T.-W. Lee, and H. Sawada, Eds. Springer, Sep 2007, ch. 12, pp. 339–364.

[103] G.-J. Jang and T.-W. Lee, "A maximum likelihood approach to single-channel source separation," *Journal of Machine Learning Research*, vol. 4, pp. 1365–1392, Dec 2003.

[104] G.-J. Jang and T.-W. Lee, "A probabilistic approach to single channel source separation," in *Neural Information Processing Systems, Advances in (NIPS)*, 2003.

[105] G.-J. Jang, T.-W. Lee, and Y.-H. Oh, "Blind separation of single channel mixture using ica basis function," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, Dec 2001, pp. 595–600.

[106] G.-J. Jang, T.-W. Lee, and Y.-H. Oh, "Single-channel signal separation using time-domain basis functions," *Signal Processing Letters, IEEE*, vol. 10, no. 6, pp. 168–171, Jun 2003.

[107] A. Jourjine, S. Rickard, and Özgür Yilmaz, "Blind separation of disjoint orthogonal signals: demixing n sources from 2 mixtures," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 5, Jun 2000, pp. 2985–2988.

[108] M. Juvela, K. Lehtinen, and P. Paatero, "The use of positive matrix factorization in the analysis of molecular line spectra," *Royal Astronomical Society, Monthly Notices of the*, vol. 280, no. 2, pp. 616–626, 1996.

[109] W. Kan, Z. Nanning, and L. Weixiang, "Natural image matting with nonnegative matrix factorization," in *Image Processing, IEEE International Conference on (ICIP)*, vol. 2, Sep 2005, pp. 1186–1189.

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

[111] E. Kim, P. K. Hopke, P. Paatero, and E. S. Edgerton, "Incorporation of parametric factors into multilinear receptor model studies of atlanta aerosol," *Atmospheric Environment*, vol. 37, no. 36, pp. 5009–5021, 2003.

[112] H. Kim and H. Park, "Discriminant analysis using nonnegative matrix factorization for nonparametric multiclass classification," in *Granular Computer, IEEE International Conference on*, May 2006, pp. 182–187.

[113] H. Kim and H. Park, "Sparse non-negative matrix factorization via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.

[114] M. Kim and S. Choi, "Monaural music source separation: Nonnegativity, sparseness, and shift-invariance," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, ser. Lecture Notes in Computer Science (LNCS), vol. 3889. Springer, Apr 2006, pp. 617–624.

[115] M. Kim and S. Choi, "On spectral basis selection for single channel polyphonic music separation," in *Artificial Neural Networks, International Conference on (ICANN)*, ser. Lecture Notes in Computer Science (LNCS), vol. 3697. Springer, Sep 2005, pp. 157–162.

[116] Y.-D. Kim and S. Choi, "Nonnegative tucker decomposition," in *Computer Vision and Pattern Recognition, IEEE Conference on (CVPR)*, Jun 2007, pp. 1–8.

[117] Y.-D. Kim and S. Choi, "A method of initialization for nonnegative matrix factorization," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 2, Apr 2007, pp. 537–540.

[118] Y.-D. Kim, A. Cichocki, and S. Choi, "Nonnegative tucker decomposition with alpha-divergence," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, Mar 2008, pp. 1829–1832.

[119] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," Sandia National Laboratories,Albuquerque, New Mexico and Livermore, California, Tech. Rep., Nov 2007.

[120] R. Kompass, "A generalized divergence measure for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 3, pp. 780–791, 2007.

[121] K. Kreutz-Delgado, B. Rao, and K. Engan, "Novel algorithms for learning overcomplete dictionaries," in *Signals, Systems, and Computers, Asilomar Conference on*, vol. 2, 1999, pp. 971–975.

[122] T. Kristjansson, J. Hershey, P. Olsen, S. Rennie, and R. Gopinath, "Superhuman multi-talker speech recognition: The IBM 2006 speech separation challenge system," in *Spoken Language Processing, ISCA International Conference on (INTERSPEECH)*, 2006, pp. 97–100.

[123] H. Laurberg, "Uniqueness of non-negative matrix factorization," in *Statistical Signal Processing, IEEE Workshop on*, Aug 2007, pp. 44–48.

[124] H. Laurberg, M. G. Christensen, M. D. Plumblery, L. K. Hansen, and S. H. Jensen, "Theorems on positive data: On the uniqueness of NMF," *Computational Intelligence and Neuroscience*, 2008.

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

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

[127] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Neural Information Processing Systems, Advances in (NIPS)*, 2000, pp. 556–562.

[128] D. D. Lee and S. H. Seung, "Unsupervised learning by convex and conic coding," in *Neural Information Processing Systems, Advances in (NIPS)*, 1996, pp. 515–521.

[129] E. Lee, C. K. Chan, and P. Paatero, "Application of positive matrix factorization in source apportionment of particulate pollutants in Hong Kong," *Atmospheric Environment*, vol. 33, no. 19, pp. 3201–3212, 1999.

[130] H. Lee, A. Cichocki, and S. Choi, "Nonnegative matrix factorization for motor imagery eeg classification," in *Artificial Neural Networks, International Conference on (ICANN)*, ser. Lecture Notes in Computer Science (LNCS), vol. 4132. Springer, Sep 2006, pp. 250–259.

[131] H. Lee, Y.-D. Kim, A. Cichocki, and S. Choi, "Nonnegative tensor factorization for continuous eeg classification," *Neural Systems, International Journal of*, 2007.

[132] J. S. Lee, D. D. Lee, S. Choi, and D. S. Lee, "Application of non-negative matrix factorization to dynamic positron emission tomography," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, Dec 2001, pp. 629–632.

[133] J. S. Lee, D. Lee, S. Choi, K. S. Park, and D. S. Lee, "Non-negative matrix factorization of dynamic images in nuclear medicine," in *Nuclear Science Symposium Conference Record*, vol. 4, 2001, pp. 2027–2030.

[134] M. S. Lewicki, "Efficient coding of natural sound," *Nature Neuroscience*, vol. 5, no. 4, pp. 356–363, Apr 2002.

[135] M. S. Lewicki and T. J. Sejnowski, "Learning overcomplete representations," *Neural Computation*, vol. 12, no. 2, pp. 337–365, 2000.

[136] M. S. Lewicki and T. J. Sejnowski, "Learning nonlinear overcomplete representations for efficient coding," in *Neural Information Processing Systems, Advances in (NIPS)*, 1998, pp. 556–562.

[137] H. Li, T. Adali, W. Wang, D. Emge, and A. Cichocki, "Non-negative matrix factorization with orthogonality constraints and its application to raman spectroscopy," *VLSI Signal Processing, Journal of*, vol. 48, pp. 83–97, Aug 2007.

[138] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. Cheng, "Learning spatially localized, parts-based representation," in *Computer Vision and Pattern Recognition, Proceedings of the IEEE Computer Society Conference on*, vol. 1, Dec 2001, pp. 207–212.

[139] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Data Mining, IEEE/WIC/ACM International Conference on (ICDM)*, Dec 2006, pp. 362–371.

[140] Y. Li and A. Cichocki, "Non-negative matrix factorization and its application in blind sparse source separation with less sensors than sources," in *Theoretical Electrical Engineering, International Symposium on (ISTET)*, 2003, pp. 285–288.

[141] C.-J. Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, pp. 2756–2779, 2007.

[142] C.-J. Lin, "On the convergence of multiplicative update algorithms for non-negative matrix factorization," *Neural Networks, IEEE Transactions on*, vol. 18, pp. 1589–1596, 2007.

[143] W. Liu and N. Zheng, "Non-negative matrix factorization based methods for object recognition," *Pattern Recognition Letters*, vol. 25, no. 8, pp. 893–897, Jun 2004.

[144] W. Liu, N. Zheng, and X. Lu, "Non-negative matrix factorization for visual coding," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 3, Apr 2003, pp. 293–296.

[145] W. Liu, N. Zheng, and X. Li, "Relative gradient speeding up additive updates for nonnegative matrix factorization," in *New Aspects in Neurocomputing: European Symposium on Artificial Neural Networks*, ser. Neurocomputing, vol. 57. Elsevier, Mar 2004, pp. 493–499.

[146] J. Lu and L. Wu, "Technical details and programming guide for a general two-way positive matrix factorization algorithm," *Chemometrics, Journal of*, vol. 18, no. 12, pp. 519–525, May 2005.

[147] M. Mørup, M. N. Schmidt, and L. K. Hansen, "Invariant sparse coding of image and music data," Technical University of Denmark, Tech. Rep., 2006.

[148] R. M. Neal, "Probabilistic inference using markov chain monte carlo methods," Dept. of Computer Science, University of Toronto, Tech. Rep., 1993.

[149] M. Novak and R. Mammone, "Improvement of non-negative matrix factorization based language model using exponential models," in *Automatic Speech Recognition and Understanding (ASRU)*, Dec 2001, pp. 190–193.

[150] M. Novak and R. Mammone, "Use of non-negative matrix factorization for language model adaptation in a lecture transcription task," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 1, May 2001, pp. 541–544.

[151] M. F. Ochs, R. S. Stoyanova, F. Arias-Mendoza, and T. R. Brown, "A new method for spectral decomposition using a bilinear bayesian approach," *Journal of Magnetic Resonance*, vol. 137, pp. 161–176, 1999.

[152] P. D. O'Grady and B. A. Pearlmutter, "Discovering convolutive speech phones using sparseness and non-negativity constraints," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, ser. Lecture Notes in Computer Science (LNCS), vol. 4666. Springer, Sep 2007, pp. 520–527.

[153] E. Oja and M. Plumbley, "Blind separation of positive sources using non-negative pca," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, 2003, pp. 11–16.

[154] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.

[155] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, Aug 2004.

[156] A. Ozerov, "Adaptation de modèles statistiques pour la séparation de sources mono-capteur application à la séparation voix / musique dans les chansons," Ph.D. dissertation, University of Rennes, 2006.

[157] A. Ozerov, P. Philippe, R. Gribonval, and F. Bimbot, "One microphone singing voice separation using source-adapted models," in *Applications of Signal Processing to Audio and Acoustics, IEEE Workshop on*, Oct 2005, pp. 90–93.

[158] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of bayesian models for single channel source separation and its application to voice / music separation in popular songs," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1564–1578, Jul 2007.

[159] P. Paatero, "Least squares formulation of robust non-negative factor analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 37, no. 1, pp. 23–35, 1997.

[160] P. Paatero, "A weighted non-negative least squares algorithm for three-way parafac factor analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 38, no. 2, pp. 223–242, 1997.

[161] P. Paatero, "The multilinear engine: a table-driven least squares program for solving multilinear problems, including the n way parallel factor analysis model," *Computational and Graphical Statistics, Journal of*, vol. 8, no. 4, pp. 854–888, 1999.

[162] P. Paatero and U. Tapper, "Positive matrix factorization: A nonnegative factor model with optimal utilization of error-estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, Jun 1994.

[163] R. M. Parry and I. Essa, "Incorporating phase information for source separation via spectrogram factorization," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 2, 2007, pp. 664–664.

[164] R. M. Parry and I. Essa, "Phase-aware non-negative spectrogram factorization," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, ser. Lecture Notes in Computer Science (LNCS), vol. 4666. Springer, Sep 2007, pp. 536–543.

[165] A. Pascual-Montano, J. Carazo, K. Kochi, D. Lehmann, and R. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 3, pp. 403–415, Mar 2006.

[166] V. P. Pauca, R. J. Plemmons, M. Giffin, and K. M. Hamada, "Unmixing spectral data for space objects using low-rank non-negative matrix factorization," in *AMOS Technical Conference*, Sep 2004.

[167] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, "Text mining using non-negative matrix factorizations," in *Data Mining, Proceedings of SIAM International Conference on*, Apr 2004.

[168] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear Algebra and its Applications*, vol. 416, no. 1, pp. 29–47, Jul 2006.

[169] B. A. Pearlmutter and R. K. Olsson, "Linear program differentiation for single-channel speech separation," *Imaging Systems and Technology, International Journal of*, vol. 15, pp. 18–33, 2005.

[170] M. D. Plumbley, "Conditions for nonnegative independent component analysis," *Signal Processing Letters, IEEE*, vol. 9, no. 6, pp. 177–80, Jun 2002.

[171] M. D. Plumbley, "Algorithms for nonnegative independent component analysis," *Neural Networks, IEEE Transactions on*, vol. 14, no. 3, pp. 534–543, May 2003.

[172] M. D. Plumbley and E. Oja, "A "nonnegative pca" algorithm for independent component analysis," *Neural Networks, IEEE Transactions on*, vol. 15, no. 1, pp. 66–76, Jan 2004.

[173] N. H. Pontoppidan and M. Dyrholm, "Fast monaural separation of speech," in *Signal Processing in Audio Recording and Reproduction, AES International Conference on*, May 2003.

[174] S. A. Raczyński, N. Ono, and S. Sagayama, "Multipitch analysis with harmonic nonnegative matrix approximation," in *Music Information Retrieval, International Conference on (ISMIR)*, Sep 2007.

[175] M. H. Radfar and R. M. Dansereau, "Single-channel speech separation using soft mask filtering," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 8, pp. 2299–2310, Nov 2007.

[176] M. H. Radfar and R. M. Dansereau, "Single channel speech separation using minimum mean square error estimation of sources' log spectra," in *Machine Learning for Signal Processing, IEEE International Workshop on*, Aug 2007, pp. 128–132.

[177] M. H. Radfar and R. M. Dansereau, "Single channel speech separation using maximum a posteriori estimation," in *Spoken Language Processing, ISCA International Conference on (INTERSPEECH)*, 2007.

[178] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, "A joint identification-separation technique for single channel speech separation," in *Digital Signal Processing Workshop, IEEE*, Sep 2006, pp. 76–81.

[179] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, "Performance evaluation of three features for model-based single channel speech separation problem," in *Spoken Language Processing, ISCA International Conference on (INTERSPEECH)*, 2006.

[180] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, "A novel low complexity vq-based single channel speech separation technique," in *Signal Processing and Information Technology, IEEE International Symposium on (ISSPIT)*, Aug 2006, pp. 572–577.

[181] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, "On the choice of window size in model-based single channel speech separation," in *Electrical and Computer Engineering, Canadian Conference on*, May 2006, pp. 298–301.

[182] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, "A maximum likelihood estimation of vocal-tract-related filter characteristics for single channel speech separation," *Audio, Speech, and Music Processing, EURASIP Journal on*, 2007.

[183] B. Raj and P. Smaragdis, "Latent variable decomposition of spectrograms for single channel speaker separation," in *Applications of Signal Processing to Audio and Acoustics, IEEE Workshop on*, Oct 2005, pp. 17–20.

[184] B. Raj, R. Singh, and P. Smaragdis, "Recognizing speech from simultaneous speakers," in *Speech Communication and Technology, European Conference on (EUROSPEECH)*, 2005, pp. 3317–3320.

[185] Z. Ramadan, B. Eickhout, X.-H. Song, L. Buydens, and P. K. Hopke, "Comparison of positive matrix factorization and multilinear engine for the source apportionment of particulate pollutants," *Chemometrics and Intelligent Laboratory Systems*, vol. 66, no. 1, pp. 15–28, 2003.

[186] R. Ramanath, R. G. Kuehni, W. E. Snyder, and D. Hinks, "Spectral spaces and color spaces," *Color Research and Application*, vol. 29, no. 1, pp. 29–37, Dec 2003.

[187] N. Rao, S. J. Shepherd, and D. Yao, "Extracting characteristic patterns from genome-wide expression data by non-negative matrix factorization," in *Computational Systems Bioinformatics Conference (CSB)*, Aug 2004, pp. 570–571.

[188] S. Rebhan, J. Eggert, H.-M. Groß, and E. Körner, "Sparse and transformation-invariant hierarchical nmf," in *Artificial Neural Networks, International Conference on (ICANN)*, ser. Lecture Notes in Computer Science (LNCS), vol. 4668. Springer, Sep 2007, pp. 894–903.

[189] A. M. Reddy and B. Raj, "Soft mask methods for single-channel speaker separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 6, pp. 1766–1776, Aug 2007.

[190] A. M. Reddy and B. Raj, "A minimum mean squared error estimator for single channel speaker separation," in *Spoken Language Processing, ISCA International Conference on (INTERSPEECH)*, 2004, pp. 2445–2448.

[191] S. J. Rennie, "Graphical models for robust speech recognition in adverse environments," Ph.D. dissertation, University of Toronto, 2008.

[192] S. J. Rennie, J. R. Hershey, and P. A. Olsen, "Efficient model-based speech separation and denoising using non-negative subspace analysis," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, 2008, pp. 1833–1836.

[193] M. J. Reyes-Gomez, D. P. W. Ellis, and N. Jojic, "Multiband audio modeling for single channel acoustic source separation," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, 2004.

[194] S. Rickard and F. Dietrich, "Doa estimation of many w-disjoint orthogonal sources from two mixtures using duet," in *Statistical Signal and Array Processing, IEEE Workshop on*, Aug 2002, pp. 311–314.

[195] S. Rickard and Özgür Yilmaz, "On the approximate w-disjoint orthogonality of speech," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, vol. 52, no. 7, Jul 2002, pp. 1830–1847.

[196] C. P. Robert, A. Doucet, and S. J. Godsill, "Marginal MAP estimation using Markov chain Monte Carlo," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 3, Mar 1999, pp. 1753–1756.

[197] S. T. Roweis, "Factorial models and refiltering for speech separation and denoising," in *Speech Communication and Technology, European Conference on (EUROSPEECH)*, 2003, pp. 1009–12.

[198] S. T. Roweis, "One microphone source separation," in *Neural Information Processing Systems, Advances in (NIPS)*, 2000, pp. 793–799.

[199] P. Sajda, S. Du, and L. Parra, "Recovery of constituent spectra using non-negative matrix factorization." in *Wavelets: Applications in Signal and Image Processing, Proceedings of SPIE*, vol. 5207, 2003, pp. 321–331.

[200] P. Sajda, S. Du, T. R. Brown, R. Stoyanova, D. C. Shungu, X. Mao, and L. C. Parra, "Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain," *Medical Imaging, IEEE Transactions on*, vol. 23, no. 12, pp. 1453–1465, Dec 2004.

[201] F. Sha and L. Saul, "Real-time pitch determination of one or more voices by nonnegative matrix factorization," in *Neural Information Processing Systems, Advances in (NIPS)*, 2005.

[202] F. Shahnaz, M. W. Berry, V. P. Pauca, and R. J. Plemmons, "Document clustering using nonnegative matrix factorization," *Information Processing and Management*, vol. 42, no. 2, pp. 373–386, 2006.

[203] M. V. Shashanka, B. Raj, and P. Smaragdis, "Sparse overcomplete decomposition for single channel speaker separation," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 2, Apr 2007, pp. 641–644.

[204] P. Smaragdis and J. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Applications of Signal Processing to Audio and Acoustics, IEEE Workshop on*, 2003, pp. 177–180.

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

[206] P. Smaragdis, "Probabilistic decompositions of spectra for sound separation," in *Blind Speech Separation*, S. Makino, T.-W. Lee, and H. Sawada, Eds. Springer, Sep 2007, ch. 13, pp. 365–386.

[207] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," in *Statistical and Perceptual Audio Processing, Workshop on (SAPA)*, Oct 2004.

[208] P. Smaragdis, "Non-negative matrix factor deconvolution; extraction of multiple sound sourses from monophonic inputs," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, ser. Lecture Notes in Computer Science (LNCS), vol. 3195. Springer, Sep 2004, pp. 494–499.

[209] S. Sra and I. S. Dhillon, "Nonnegative matrix approximation: Algorithms and applications," University of Texas at Austin, Tech. Rep., Jun 2006.

[210] S. Srinivasan, J. Samuelsson, and W. Kleijn, "Codebook-based bayesian speech enhancement," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 1, Mar 2005, pp. 1077–1080.

[211] S. Srinivasan, J. Samuelsson, and W. Kleijn, "Codebook driven short-term predictor parameter estimation for speech enhancement," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 1, pp. 163–176, Jan 2006.

[212] S. Srinivasan, N. Roman, and D. Wang, "Binary and ratio time-frequency masks for robust speech recognition," *Speech Communication*, vol. 48, pp. 1486–1501, 2006.

[213] K. Stadlthanner, F. J. Theis, C. G. Puntonet, and E. W. Lang, "Extended sparse nonnegative matrix factorization," in *Artificial Neural Networks, International Workshop on (IWANN)*, vol. 3512, 2005, pp. 249–256.

[214] F. J. Theis, K. Stadlthanner, and T. Tanaka, "First results on uniqueness of sparse non-negative matrix factorization," in *European Signal Processing Conference, Proceedings of (EUSIPCO)*, 2005.

[215] T. Tolonen, "Methods for separation of harmonic sound sources using sinusoidal modeling," in *Audio Engineering Society (AES) Convention*, May 1999.

[216] S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita, "Dimensionality reduction using non-negative matrix factorization for information retrieval," in *Systems, Man, and Cybernetics, IEEE International Conference on*, vol. 2, 2001, pp. 960–965.

[217] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279—311, 1966.

[218] E. Vincent and M. D. Plumbley, "Single-channel mixture decomposition using bayesian harmonic models," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA)*, ser. Lecture Notes in Computer Science, vol. 3889, Feb 2006, pp. 722–730.

[219] T. Virtanen and A. Klapuri, "Separation of harmonic sound sources using sinusoidal modeling," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on (ICASSP)*, vol. 2, 2000, p. 765.

[220] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," in *International Computer Music Conference, ICMC*, 2003.

[221] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 1066–1074, Mar 2007.

[222] T. Virtanen, "Separation of sound sources by convolutive sparse coding," in *Statistical and Perceptual Audio Processing, Workshop on (SAPA)*, Oct 2004.

[223] T. Virtanen, "Speech recognition using factorial hidden markov models for separation in the feature space," in *Spoken Language Processing, ISCA International Conference on (INTERSPEECH)*, 2006.

[224] T. Virtanen, "Sound source separation in monaural music signals," Ph.D. dissertation, Tampere University of Technology, 2006.

[225] B. Wang and M. D. Plumbley, "Musical audio stream separation by non-negative matrix factorization," in *DMRN Summer Conference, Glasgow, Proceedings of the*, Jul 2005.

[226] B. Wang and M. D. Plumbley, "Investigating single-channel audio source separation methods based on non-negative matrix factorization," in *ICA Research Network International Workshop, Proceedings of*, Sep 2006, pp. 17–20.

[227] G. Wang, A. V. Kossenkov, and M. F. Ochs, "LS-NMF: A modified non-negative matrix factorization algorithm utilizing uncertainty estimates," *BMC Bioinformatics*, vol. 7, no. 175, Mar 2006.

[228] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Fisher non-negative matrix factorization for learning local features," in *Computer Vision, Asian Conference on*, Jan 2004.

[229] R. J. Weiss and D. P. W. Ellis, "Speech separation using speaker-adapted eigenvoice speech models," *Computer Speech and Language*, Mar 2008.

[230] M. Welling and M. Weber, "Positive tensor factorization," *Pattern Recognition Letters*, vol. 22, no. 12, pp. 1255–1261, 2001.

[231] H. Wersing, J. Eggert, and E. Körner, "Sparse coding with invariance constraints," in *Artificial Neural Networks, International Conference on (ICANN)*, ser. Lecture Notes in Computer Science (LNCS), vol. 2714. Springer, Jun 2003, pp. 385–392.

[232] S. Wild, J. Curry, and A. Dougherty, "Improving non-negative matrix factorizations through structured initialization," *Pattern Recognition*, vol. 37, no. 11, pp. 2217–2232, Nov 2004.

[233] O. Winther and K. B. Petersen, "Bayesian independent component analysis: Variational methods and non-negative decompositions," *Digital Signal Processing*, vol. 17, no. 5, pp. 858–872, 2007.

[234] Y.-L. Xie, P. K. Hopke, and P. Paatero, "Positive matrix factorization applied to a curve resolution problem," *Chemometrics, Journal of*, vol. 12, no. 6, pp. 357–364, 1998.

[235] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Research and Development in Information Retrieval, Proceedings of the International SIGIR Conference on*, 2003, pp. 267–273.

[236] F. Yin, J. Wang, C. Guo, W. Liu, N. Zheng, and X. Li, "Nonnegative matrix factorization for eeg signal classification," in *Advances in Neural Networks (ISSN)*, ser. Lecture Notes in Computer Science (LNCS), vol. 3174. Springer, 2001, pp. 470–475.

[237] S. S. Young, P. Fogel, and D. Hawkins, "Clustering scotch whiskies using non-negative matrix factorization," *Joint Newsletter for the Section on Physical and Engineering Sciences and the Quality and Productivity Section of the American Statistical Association*, vol. 14, no. 1, pp. 11–13, Jun 2006.

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

[239] R. Zdunek and A. Cichocki, "Non-negative matrix factorization with quasi-newton optimization," in *Artificial Intelligence and Soft Computing, International Conference on (ICAISC)*, vol. 4029, Jun 2006, pp. 870–879.

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

[241] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with quadratic programming," *Neurocomputing*, 2007.

[242] D. Zhang, Z.-H. Zhou, and S. Chen, "Non-negative matrix factorization on kernels," in *Artificial Intelligence, Pacific Rim International Conference on*, ser. Lecture Notes in Artificial Intelligence (LNAI), vol. 4099. Springer, Aug 2006, pp. 404–412.

[243] J. Zhang, L. Wei, Q. Miao, and Y. Wang, "Image fusion based on non-negative matrix factorization," in *Image Processing, IEEE International Conference on (ICIP)*, vol. 2, Oct 2004, pp. 973–976.

[A] Mikkel N. Schmidt and Morten Mørup, "Non-negative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation," in *Independent Component Analysis and Blind Signal Separation, International Conference on (ICA), Lecture Notes in Computer Science (LNCS)*, Springer, vol. 3889, pp. 700–707, Apr. 2006.

[B] Mikkel N. Schmidt and Rasmus K. Olsson, "Single-Channel Speech Separation using Sparse Non-Negative Matrix Factorization," in *Spoken Language Processing, ICSLP International Conference on (INTERSPEECH)*, Sep. 2006.

[C] Mikkel N. Schmidt, Jan Larsen, and Fu-Tien Hsiao, "Wind Noise Reduction using Non-negative Sparse Coding," in *Machine Learning for Signal Processing, IEEE International Workshop on (MLSP)*, pp. 431–436, Aug. 2007.

[D] Mikkel N. Schmidt and Rasmus K. Olsson, "Linear Regression on Sparse Features for Single-Channel Speech Separation," in *Applications of Signal Processing to Audio and Acoustics, IEEE Workshop on (WASPAA)*, Oct. 2007.

[E] Mikkel N. Schmidt and Hans Laurberg, "Non-negative Matrix Factorization with Gaussian Process Priors," in *Computational Intelligence and Neuroscience*, May 2008.