Doctoral Thesis

Source-Filter Model Based Single Channel Speech Separation

Michael Stark

Signal Processing and Speech Communication Laboratory
Faculty of Electrical and Information Engineering
Graz University of Technology, Austria

Advisors:
Prof. Gernot Kubin, Graz University of Technology, Austria
Dr. Franz Pernkopf, Graz University of Technology, Austria

Thesis Examiners:
Prof. Gernot Kubin, Graz University of Technology, Austria
Assoc. Prof. Ioannis Stylianou, University of Crete, Greece

Graz, May 2010

Abstract

In a natural acoustic environment, multiple sources are usually active at the same time. The task of source separation is the estimation of individual source signals from this complex mixture. The challenge of single channel source separation (SCSS) is to recover more than one source from a single observation. Basically, SCSS can be divided in methods that try to mimic the human auditory system and model-based methods, which find a probabilistic representation of the individual sources and employ this prior knowledge for inference.

This thesis presents several strategies for the separation of two speech utterances mixed into a single channel and is structured in four parts: The first part reviews factorial models in model-based SCSS and introduces the soft-binary mask for signal reconstruction. This mask shows improved performance compared to the soft and the binary masks in automatic speech recognition (ASR) experiments.

The second part addresses the problem of computational complexity in factorial models, which limits its application for online processing. We introduce the fast beam search and the iterated conditional modes (ICM) approximation techniques. They reduce the computational complexity in factorial models by up to two orders of magnitude while maintaining the separation performance. Moreover, there is strong evidence that the ICM algorithm breaks the factorial structure entirely. Consequently, this leads to a linear complexity relationship in the number of hidden states instead of a factorial one.

The third part deals with arbitrary mixing levels in factorial models by explicitly modeling the gain for each speech segment, which results in a shape-gain model. Several strategies for parallel estimation of gain and shape are successfully evaluated.

Finally, the last part integrates the speech model in model-based systems. This results in a source-filter representation, where the source signal can be linked to the excitation signal of the vocal folds and the filter accounts for the vocal-tract shaping. Our final separation algorithm combines the shape-gain with the source-filter model, reflecting the complete standard speech production model. All presented algorithms are compared to state-of-the-art algorithms and evaluated in both, the target-to-masker ratio and the word error rate of an ASR system and show improvements beyond the state-of-the-art.

Kurzfassung

In einer natürlichen akustischen Umgebung sind meist mehrere Quellen zur gleichen Zeit aktiv. Das Ziel der Quellentrennung ist die Schätzung der Einzelsignale aus dieser komplexen Mixtur. Die Herausforderung der einkanaligen Quellentrennung ist die Trennung mehrerer Quellen an Hand einer einzigen Beobachtung. Grundsätzlich wird die einkanalige Quellentrennung in Methoden, die das menschliche Gehör imitieren, und in modellbasierte Methoden unterteilt. Modellbasierte Methoden können die Wahrscheinlichkeitsverteilung der einzelnen Quellen während des Trainings erlernen und vereinen diese zu einem faktoriellen Modell, um die Quellen zu trennen.

Das Ziel dieser Arbeit ist die Entwicklung von Strategien zur Trennung zweier Sprachsignale und sie ist in vier Teile unterteilt: Der erste Teil beschreibt faktorielle Modelle und führt die soft-binary mask zur Signaltrennung ein. Diese Maske zeigt sehr gute Resultate in Spracherkennungstests.

Der zweite Teil behandelt die Rechenkomplexität von faktoriellen Modellen, die den Echtzeiteinsatz dieser Algorithmen limitiert. Wir stellen mit dem fast beam search und dem iterated conditional modes (ICM) Algorithmus zwei Näherungsverfahren vor. Beide Ansätze reduzieren den Rechenaufwand um zwei Größenordnungen bei nahezu gleichen Resultaten. Es gibt Hinweise darauf, dass der ICM Algorithmus den Faktorgraphen aufbrechen kann und dadurch die Komplexität soweit reduziert, dass sie nur noch linear in der Anzahl der verborgenen Zustände ist.

Im dritten Teil wird das Problem der Schätzung des Signalmischverhältnisses behandelt. Zur Lösung schlagen wir die separate Modellierung des Spektralverlaufs und der Verstärkung vor. Dies führt zu einem Shape-Gain Faktorgraphen. Für diesen werden unterschiedliche Strukturen und Methoden zur Schätzung der Verstärkung hinsichtlich ihrer Leistungsfähigkeit erfolgreich evaluiert.

Im letzten Teil dieser Arbeit werden sprachspezifische Merkmale in das Modell integriert, indem das Sprachsignal als Quelle-Filter-Modell dargestellt wird. Das Anregungssignal entspricht der Schwingung der Stimmbänder und das Filter der Signalformung durch den Mund- und Rachenraum. Zur Quellentrennung werden Quelle und Filter im Faktorgraphen durch separate Zufallsvariablen dargestellt. Die Kombination des Shape-Gain Ansatzes mit dem Quelle-Filter Modell führt schließlich zum vollständigen Standardmodell der Spracherzeugung. Die Qualität der Trennung wird mittels des Target-to-Masker Verhältnisses und der Wortfehlerrate eines Spracherkenners bestimmt, wobei sich Verbesserungen gegenüber dem Stand der Technik zeigen.

TU Graz
Graz University of Technology
Senat

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

EIDESSTATTLICHE ERKLÄRUNG

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

Graz, am …………………………… ………………………………………………..
(Unterschrift)

Englische Fassung:

STATUTORY DECLARATION

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

…………………………… ………………………………………………..
date (signature)

Acknowledgments

This work was carried out at the Signal Processing and Speech Communication Laboratory, Graz University of Technology, during 2006-2010. I really want to thank all members of the research group for creating such an inspiring atmosphere to work. I wish to express my deepest gratitude to my supervisor Professor Gernot Kubin for guiding me during my time here.

I owe a special thanks to Franz Pernkopf, always highly motivated, competent, and open for discussions whenever needed. Thanks for the fun with you and for proofreading my thesis, which increased the quality significantly.

In different ways it was an enriching experience to spend time at the Computer Science Department at the University of Crete. In this respect, I especially want to thank Professor Yannis Stylianou for being my supervisor in this period of time and for participating in my thesis examination committee.

This work was carried out in the context of COAST-ROBUST. The founding by the Austrian KNet Program, ZID Zentrum für Innovation und Technology, Vienna, the Steirische Wirtschaftsförderungsgesellschaft mbH, and the Land Steiermark is gratefully acknowledged.

My warmest thanks go to my dear wife Brigitte for her support, love and understanding during all the time it took me to put the work together.

Michael Stark
Graz, Austria, May 2010

Contents

1 Introduction **3**

1.1 Scope of the Thesis 4

1.2 Related PhD Theses 6

1.2.1 M.H. Radfar: Single Channel Speech Separation 6

1.2.2 M. Reyes-Gomez: Statistical Graphical Models for Scene Analysis, Source Separation and other Audio Applications 8

1.2.3 S.J. Rennie: Graphical Models for Robust Speech Recognition in Adverse Environments 9

1.3 Organization 9

1.4 Applications 10

1.5 Experimental Setup 11

2 An Overview of Work in Single-Channel Speech Separation **15**

2.1 Computational Auditory Scene Analysis (CASA) 15

2.1.1 Temporal Binding and Oscillatory Correlation 16

2.1.2 Cortronic Network 16

2.2 SCSS based on Basis Decomposition 17

2.2.1 Independent Component Analysis 17

2.2.2 Non-Negative Matrix Factorization 19

2.2.3 Latent Variable Decomposition 21

2.3 Separation based on Modulation Frequency Analysis 21

2.4 Evaluation Methods 22

2.4.1 Objective Quality Measures 22

2.4.2 Quality Evaluation by Subjective Tests 25

2.5 Conclusion 26

3 Model-Based Single Channel Source Separation 27
3.1 Introduction . . . 27
3.2 Inference in Factorial Models . . . 29
3.2.1 Observation Likelihood Computation . . . 30
3.3 Observation Models for Factorial Models . . . 31
3.3.1 Additive Model . . . 32
3.3.2 Mixture-Maximization Model . . . 32
3.3.3 Algonquin Model . . . 33
3.4 Signal Reconstruction . . . 33
3.5 Experimental Results . . . 34
3.6 Conclusion . . . 40

4 Computational Complexity in Factorial Models 41
4.1 Introduction . . . 41
4.2 Full Observation Likelihood Computation (FS) . . . 42
4.3 Fast Likelihood Estimation (FLE) . . . 42
4.4 Beam Search Technique (BS) . . . 43
4.5 Gibbs Sampling (GS) . . . 45
4.6 Computational Complexity . . . 48
4.6.1 Performance and Cost Function . . . 48
4.7 Experiments and Results . . . 49
4.8 Conclusion . . . 53

5 Gain And Shape Modeling For Source Separation 55
5.1 Introduction . . . 55
5.2 Gain Estimation . . . 57
5.2.1 Maximum-Likelihood Gain Estimation . . . 57
5.2.2 Nonlinear Gain Estimation . . . 59
5.2.3 Projection based Gain Estimation . . . 60
5.2.4 Nonnegative Least Squares based Gain Estimation . . . 60
5.2.5 Nonnegative Matrix Factorization based Gain Estimation . . . 61
5.2.6 Auditory Motivated Gain Estimation . . . 61
5.3 Combined Shape and Gain Estimation . . . 62
5.3.1 Shape-Gain Decoder . . . 62
5.3.2 Gain-Shape Decoder . . . 62
5.3.3 Independent Speaker Decoder . . . 62
5.4 Experiments and Results . . . 64
5.4.1 Supervised Gain Estimation . . . 64
5.4.2 Shape-Gain Single Channel Source Separation . . . 68
5.5 Conclusion . . . 72

6 Source Specific Characteristic for SCSS 73

6.1 Introduction . . . 73

6.1.1 Source- and Model-Driven Approach . . . 74

6.2 Sequential Source-Filter based SCSS . . . 75

6.2.1 Synthetic Excitation Modeling . . . 76

6.2.2 Direct Statistical Excitation Modeling . . . 81

6.3 Parallel Source-Filter based SCSS . . . 86

6.3.1 Observation Model . . . 87

6.4 Experiments and Results . . . 89

6.4.1 Synthetic Excitation Results . . . 90

6.4.2 Separation Results for the Direct Statistical Excitation Model . . . 100

6.4.3 Parallel Source-Filter Model Separation Results . . . 101

6.4.4 Sequential versus Parallel Source-Filter SCSS . . . 103

6.5 Conclusion . . . 103

7 Conclusion and Future Directions 105

7.1 Summary and Contributions . . . 105

7.2 Future Directions . . . 107

A Experiments in a Real-World Environment 109

A.1 Recording Setup . . . 109

A.2 Recording Equipment . . . 111

A.3 Results . . . 111

A.4 Conclusion . . . 113

B Automatic Speech Recognition 115

B.1 ASR Results on the *SAIL Real Life Corpus* . . . 115

B.2 ASR Results on the WSJ0 Nov92 Test Set . . . 116

B.3 Conclusion . . . 118

Bibliography . . . 121

List of Figures

1.1 Target-source separation problem with open-set interferers. 3

1.2 Co-channel source separation problem. 4

1.3 Spectrogram, orthographic transcription and time-domain signal of the sentence “lay white in x zero soon” of speaker FE1. 12

1.4 Example result plot. Same gender female (SGF), male (SGM), and different gender (DG) results are plotted separately. The labels identify different methods. 13

3.1 Source separation problem: The mixture Y is obtained by the combination of the two source signals S_i in every time step τ. 27

3.2 Source separation problem using generative models. (a) The mixture Y is obtained by the combination of the means of the two emission densities $\phi_s(z_i)$ specified by their hidden variables z_i in every time step τ. (b) Factor graph representation: The conditional dependency of the source signal on the random variable is explicitly shown by the factor nodes (filled nodes). Circles denote variable nodes. 29

3.3 Factor graph representation of the factorial model. The mixture $\mathbf{y}$ is represented by two hidden variables z_i in every time step τ. 30

3.4 Factorial-max VQ single channel separation performance. The dependency between separation performance and the used number of codewords (Q) to model speaker characteristics are investigated. 35

3.5 Factorial-max VQ separation results. In this experiment the number of components is fixed, varying just the number of iterations (iT) for model training. 36

3.6 Factorial-max VQ separation results employing a speaker specific template dictionary. The label indicates the cardinality of the template dictionary D employed for separation. 37

3.7 Ideal masks of target speaker FE1. **Top:** ideal binary mask. **Middle:** ideal soft mask. **Bottom:** ideal soft binary mask. White pixels indicate a “1” in the mask and black pixels a “0”. 38

3.8 Single channel separation results of two speakers using the ideal binary mask (iBM), ideal soft mask (iSM), and the ideal soft binary mask (iSBM).The TMR results in [dB] are illustrated for the SGF, SGM, and DG case. 39

3.9 FM-VQ single channel separation results employing the binary mask (BM), soft mask (SM), and the soft binary mask (SBM) for signal reconstruction.The TMR results in [dB] are illustrated for the SGF, SGM, and DG case. 39

4.1 Observation likelihood plane for a single frame of a female-female speech mixture. The colored lines correspond to the four initializations of the ICM4 algorithm and the numbers denote the search path sequence taken to approach the maximum. 47

4.2 Mean log-likelihood function for all state combinations averaged over all test utterances. Log-likelihoods are sorted in descending order (linear scale). The markers identify the average score of the search heuristics. 49

4.3 Mean log-likelihood function for all state combinations averaged over all test utterances. Log-likelihoods are sorted in descending order (logarithmic-scale). The markers identify the average score of the search heuristics. 50

4.4 TMR results for various setups of the fast beam search (FBS) approximation. For decoding, the factorial-max VQ structure has been employed. The first number in the labels indicates the M-best states at $\tau - 1$ to determine the N states (second number) employed for separation at time step τ. See section 4.4 for more details. 52

4.5 TMR result for factorial-max VQ observation likelihood approximations. Comparison of the baseline approximations (FBS, ICM) to their improvements (FBS-3-50, ICM4), as well as to the FS and FLE. 52

4.6 FM-HMM results using the approximation methods: beam search (BS), fast beam search (FBS), fast likelihood estimation (FLE), iterated conditional modes using Viterbi (ICM), ICM4 with Viterbi tracking (Vit-ICM4), and ICM4 with tracking (max-ICM4). The TMR is depicted separately for the SGF, SGM, and DG cases. See section 1.5 for more details. 53

5.1 Shape-Gain production model 56

5.2 Shape-gain decoder structure for the ICM4 based observation likelihood computation. The gain factor computational complexity is significantly reduced. . 63

5.3 Gain-shape decoder structure for multi-variable gain optimization methods. . 63

5.4 Gain-shape decoder structure with independent gain estimation for each mean value associated to a hidden variable. 64

5.5 Maximum-likelihood gain adjustment. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases with different mixing levels for target (t) and masker (m) speaker. 65

5.6 Nonlinear gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker. 66

5.7 Matching pursuit gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker. 66

5.8 Nonnegative least squares gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker. 67

5.9 Nonnegative matrix factorization based gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker. 67

5.10 Auditory motivated gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker. 68

5.11 Separation performance in TMR for the ML-based gain estimation. Results are shown for gain-shape based (GS-VQ), independent gain estimation (Ind-ML-GE), and shape-gain (SG-VQ) based separation of two speech signals. . . 69

5.12 Separation results in TMR using nonlinear gain estimation. Performance is plotted for different percentiles, i.e. the 10, 20, 30, 40 and 50 percent percentile. 70

5.13 Separation results in TMR using nonnegative matrix factorization based gain estimation. Two structures are investigated for the shape estimation: (i) “Ind-GE-NMF”: The gains are estimated independently. (ii) “GS-NMF-VQ”: The gain-shape structure is employed for SCSS. 71

5.14 Separation performance comparison in TMR of introduced gain estimation methods. Performance is plotted for SGF, SGM and DG case separately. . . . 71

6.1 Speech production model. 75

6.2 Blockdiagram of the separation system. 76

6.3 A factorial HMM shown as a factor graph [1]. Factor nodes are shown as shaded rectangles together with their functional description. Hidden variable nodes are shown as circles. Observed variables $\mathbf{Y}^{(\tau)}$ are absorbed into factor nodes. 77

6.4 Block diagram of the training stage of the source-filter model. 82

6.5 Block diagram of the separation algorithm. 83

6.6 (a) Mixture of excitation signals $e_{s1,s2}(t)$, (b) Spectral whitened excitation mixture $e_y(t)$, (c) Corresponding error defined as difference signal: $e_{s1,s2}(t) - e_y(t)$. 85

6.7 Source-Filter F-HMM where $Y^{(\tau)}$ is represented by four state variables $u_1^{(\tau)}$, $v_1^{(\tau)}$, $u_2^{(\tau)}$, and $v_2^{(\tau)}$. 87

6.8 Factor graph of the parallel gain-shape source-filter SCSS model, for one time step. Random variable a_{u_i} and a_{v_i} are associated to the introduced gain factor. 89

6.9 Trajectories found by the proposed multi-pitch tracker, applied on speaker MA1 (”prwkzp”) and speaker FE1 (”lwixzs”) speaking simultaneously. The overall accuracy is high, yet some parts of the trajectory of speaker 1 can not be tracked successfully. This leads to a high contribution of E_{21} and E_{10} to the overall error. The corresponding error measures on this test instance are shown in the Table at the bottom. 91

6.10 Mean and standard deviation of the TMR for the synthesized signals using pitch trajectories extracted by *RAPT* [2]. 94
6.11 Mean and standard deviation of the TMR for the BM signals using pitch trajectories extracted by *RAPT* [2]. 95
6.12 Mean and standard deviation of the TMR for the synthesized signals using SD multi-pitch trajectories. 95
6.13 Mean and standard deviation of the TMR for the BM signals using SD multi-pitch trajectories. 96
6.14 Mean and standard deviation of the TMR for the synthesized signals using GD multi-pitch trajectories. 96
6.15 Mean and standard deviation of the TMR for the BM signals using GD multi-pitch trajectories. 97
6.16 Mean and standard deviation of the TMR for the synthesized signals using SI multi-pitch trajectories. 97
6.17 Mean and standard deviation of the TMR for the BM signals using SI multi-pitch trajectories. 98
6.18 Coherence between the average E_{Total} versus average TMR for all VTF and pitch estimation methods for the reference, SD, GD, and SI pitch tracks. Results are separately plotted for (a) synthesized speech signals; (b) BM signals. 99
6.19 Mean and standard deviation of the TMR for the sequential source-filter based SCSS methods. Here different graphical models structures are investigated to estimate the fine- and coarse spectral signal parts. 100
6.20 Mean and standard deviation of the TMR for the BM signals using sequential gain-shape source-filter based SCSS method. Here different graphical model structures are investigated for gain-shape source-filter based source separation. 101
6.21 Mean and standard deviation of the TMR for the BM signals using the parallel source-filter based SCSS method. 102
6.22 Mean and standard deviation of the TMR for the BM signals using parallel gain-shape source-filter based SCSS methods. Here different graphical model structures are investigated for gain-shape source-filter based source separation. 103
6.23 Mean and standard deviation of the TMR. Comparison of sequential and parallel source-filter based SCSS algorithms. *seq-AE* and *seq-VQVQ* denote the sequential methods using multi-pitch tracking and the direct excitation signal model. *par-VQVQ-lpc* and *par-VQVQ* refer to the parallel methods using LPC and SEEVOC to find the coarse and fine spectral structure, respectively. . . . 104

A.1 Layout and dimensions of the Cocktail Party room. The ground plan is shown in the top plot and the vertical section in the bottom plot. The gap of 3.38 m indicates a window, the circle in the upper left corner represents a column (concrete), and between the column and the door their is a long bookshelf located. The whole floor is carpeted, otherwise the walls are empty. 110
A.2 Equivalent network of the speech mixture in a real environment. Symbol h_1 and h_2 model the respective transfer function from loudspeaker to microphone. 111
A.3 Mean and standard deviation of the TMR for the BM signals using the parallel gain-shape source-filter based SCSS method. Specifically, the HMM-HMM structure of section 6.3 was used for separation. The labels indicate different distances between loudspeakers and microphone. 112

List of Tables

1.1 Label of female and male speakers used for training speaker independent models. 11

1.2 Labels of speakers and file names used for testing. 12

4.1 Complexity comparison for FM-VQ using full search (FS), fast likelihood estimation (FLE), fast beam search (FBS), Gibbs sampling (GS), and ICM4. . 48

4.2 Separation results in TMR [dB] for the full search (FS), fast likelihood estimation (FLE), fast beam search (FBS), and iterated conditional modes (ICM) methods. Average TMR results with standard deviation (Std) are listed separately for the three different mixing cases. 50

4.3 Separation results in TMR [dB] for likelihood estimation, using different Markov chain Monte Carlo methods. Average TMR results with standard deviation (Std) are listed separately for the three different mixing cases. 51

6.1 Performance of F-HMM-based multi-pitch tracking for speaker dependent (SD) training. Mean and standard deviation (std) of the nine test instances of each speaker pair are shown. 92

6.2 Performance of F-HMM-based multi-pitch tracking for gender dependent (GD) training. Mean and standard deviation (std) over the 9 test instances of each speaker pair are shown. 92

6.3 Performance of F-HMM-based multi-pitch tracking for speaker independent (SI) training. Mean and standard deviation (std) over the 9 test instances of each speaker pair are shown. 93

A.1 Recording setups for experiments in the cocktail party room. The distance $d1$ refers to the distance between loudspeakers and microphone, whereas the distance between loudspeakers is $d2$. 111

B.1 Word error rate (WER) for the mixture and the separated signals using the NMF-l_0 and the factorial-max VQ method. For the factorial-max VQ three different mask for separation are employed: (i) binary mask (BM), (ii) soft binary mask (SBM), and (iii) soft mask (SM). 116

B.2 Percentage of same gender (SGM) and different gender mixtures (DGM) for each target speaker in the database. 118

B.3 Word error rate (WER) for the clean, mixed and the separated signals. For the factorial-max VQ three different masks for separation are employed: (i) binary mask (BM), (ii) soft binary mask (SBM), and (iii) soft mask (SM). For the source-filter (SF) and the gain-shape SF (GS-SF) method the SBM is used for separation. 118

List of notations and acronyms

Mathematical notational conventions

Meaning	Type face	Examples
vector or matrix	bold, regular	$\mathbf{y}, \mathbf{s}, \mathbf{S}, A$
time sequence	regular	$s(t), y(t)$
scalar	regular	a, t
stochastic variable	lowercase	$u, v, z, \mathbf{z}$
parameter (set)	greek alphabet (uppercase)	λ, Ψ, ϕ
vector or matrix transpose	prime	$\mathbf{y}', A'$
d^{th} element of vector or matrix	superscript index	$\mathbf{y}^{(i)}, A^{(i)}$

Notation

Notation	Meaning
$s(t)$	time-domain signal of a single speaker speech signal
$\mathbf{s}$	matrix, spectrogram of single speaker speech in magnitude-frequency domain using the short-time Fourier transform (STFT)
$\mathbf{S} = \log \mathbf{s}$	matrix, spectrogram of single speaker speech in logarithmic magnitude-frequency domain
$\mathbf{s}^{(\tau)}, \mathbf{S}^{(\tau)} \in \mathbb{R}^D$	τ^{th} feature vector in matrix $\mathbf{s},\ \mathbf{S}, \quad \tau \in \{1, \ldots, T\}$
$\mathbf{s}^{(\tau,d)}, \mathbf{S}^{(\tau,d)}$	the element (τ, d) of the matrix $\mathbf{s},\ \mathbf{S}$
S, Y	complex spectrogram of signals
$\angle S,\ \angle Y$	phase spectrogram of signal
$s_i(t),\ \mathbf{s}_i,\ \mathbf{S}_i$	signal of i^{th} speaker
$y(t), \mathbf{y}, \mathbf{Y}, Y$	speech mixture signal
$\mathcal{N}$	normal distribution

AMR-WB	Adaptive Multi-Rate WideBand
ASA	Auditory Scene Analysis
ASR	Automatic Speech Recognition
BM	Binary Mask
CASA	Computational Auditory Scene Analysis
CPP	Cocktail Party Problem
DG	Different Gender
EM	Expectation-Maximisation (algorithm)
FST	Finite State Transducer
GMM	Gaussian Mixture Model
GD	Gender Dependent
HMM	Hidden Markov Model
F-HMM	Factorial-HMM
FM-HMM	Factorial-Max HMM
LPC	Linear Prediction Coding
LSF	Line Sectral Frequencies
LVCSR	Large Vocabulary Continuous Speech Recognition
MAP	Maximum A-Posteriori
MDL	Minimum Description Length
ML	Maximum Likelihood
MIXMAX	Mixture Maximization
NMF	Non Negative Matrix Factorization
NN	Neural Network
RAPT	Robust Algorithm for PiTch estimation
SD	Speaker Dependent
SEEVOC	Spectral Envelope Vocoder
SGF	Same Gender Female
SGM	Same Gender Male
SBM	Soft Binary Mask
SM	Soft Mask
SI	Speaker Independent
STFT	Short Time Fourier Transform
SVM	Support Vector Machine
UD	Utterance Dependent
VQ	Vector Quantization
VTF	Vocal Tract Filter
FM-VQ	Factorial-Max VQ
WER	Word Error Rate

Chapter 1

Introduction

In a typical real-world auditory scene, more than one source is active at the same time. The computational analysis of such an auditory scene is a challenging problem, with a great practical interest. Source separation aims at dividing this source mixture of an auditory scene into its individual sources. For this purpose, a vast amount of source separation algorithms can be found in literature [3, 4]. However, the performance is only rarely and for specific tasks comparable to the human auditory system, e.g. [5]. Human listeners have the ability to identify and follow a particular source with ease in the presence of other sources [6]. This is known as the cocktail party problem (CPP), first described by Cherry [7]. Bregman [8], provided an analysis how the human auditory system performs this task of auditory scene analysis (ASA). For the computational separation of sources, multiple acoustic sensors (microphones) placed at different spatial locations are usually considered. The usage of multiple sensors enables to additionally employ the spatial dimension for separation, which usually leads to a dramatic facilitation of separation. Algorithms like Independent Component Analysis [9], Non Negative Matrix Factorization with sparseness constraints, higher-order statistics (HOS), or second-order statistics (SOS) [10] are employed for separation. Some of them additionally employ spatio-temporal cues for separation. However, in many cases, only single channel recordings are available. The objective of separation is to recover more than one source from

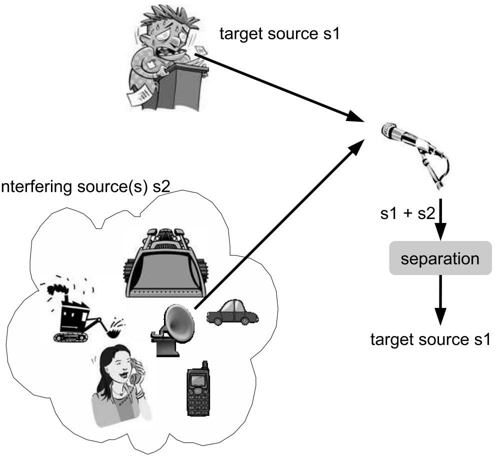


Figure 1.1: Target-source separation problem with open-set interferers.

a single observed signal. Therefore, let us assume a linear instantaneous mixture of two sources throughout the thesis:

$$y(t) = s_1(t) + s_2(t), \tag{1.1}$$

where $y(t)$ is the source mixture, $s_1(t)$ and $s_2(t)$ are the component signals, and $t = [1, \ldots, T]$ is the time index. Thus, the problem of single channel source separation (SCSS) is underdetermined, and constraints have to be defined to be able to separate the signals. Typically, source specific prior knowledge, statistical independence or constant statistical source characteristics over time are employed for separation.

In general, a source can be any auditory event, such as human speech, music, a PC fan, noise from a car passing by, or street noise. In most cases, one source is singled out as the object of interest in SCSS. This source is known as target source, while all other sources are interferences, obscuring the target source. Figure 1.1 illustrates this target-source separation problem, where only the target speech should be recovered, independent of the type and the number of interferences. As there are no restrictions regarding the interferences, this condition is called open-set scenario. Such a real-world scenario with various interferences is highly dynamic and complex. Thus, the number of component sources can change abruptly as well as their respective statistics are nonconstant over time.

In co-channel speech, utterances of two speakers are transmitted over a single communication channel [11]. Thus, one has to deal with a linear instantaneous mixture of the sources. In contrast to conversational speech, in co-channel speech the speakers are usually not aware of each other. This leads to a multitude of speech overlaps, which presents a challenging task for source separation. Figure 1.2 depicts this competing talker scenario, where two speaker are talking at the same time. In this case, source separation aims at recovering both individual sources from the mixture. The co-channel speech scenario can be considered as a special case of the general open-set speech-interference problem, where the interference is a competing speaker.

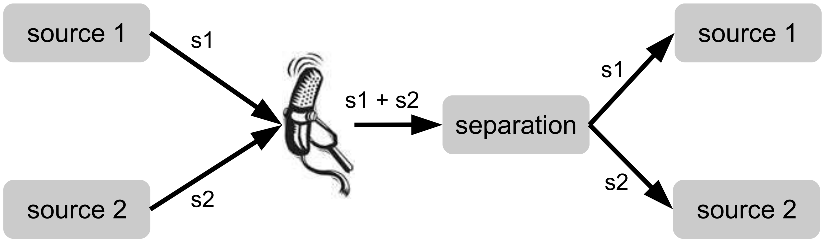


Figure 1.2: Co-channel source separation problem.

1.1 Scope of the Thesis

In the beginning of source separation research, the systems tried to mimic the human auditory system in order to solve the CPP problem. These systems took advantage of psychoacoustic cues for separation and are summarized as computational auditory scene analysis (CASA) methods. For CASA systems, the source mixture is a scene to be analyzed and organized. Therefore, low-level cues, such as continuity over time and frequency, common onset and offset, fundamental frequency, amplitude and frequency modulation, and spatial proximity are exploited to form segments, which most likely arise from a single source [3]. Problems associated with CASA systems are listed in the following:

- During simultaneous grouping in the high frequency region, the segment formation due to unresolved harmonics is non-trivial. Unresolved harmonics mainly emerge due to the frequency representation with an auditory filterbank with quasi-logarithmically center frequencies. The method of Wang et al. [12] mainly relies on resolved harmonics, which results in a low-pass filtered separated signal. Hu et al. [13] bypasses this problem partially as they converted signal components above a frequency of $\sim$ 2 kHz to an amplitude modulated signal.

- Sequential grouping: Once for every time segment source unique segments have been found, the grouping into streams is non-trivial. Recently, Shao et al. [14] proposed a clustering approach to tackle this problem.

- The separation of unvoiced speech is a challenge, since most CASA systems heavily rely on harmonicity. In [15] the separation of unvoiced speech from non-speech interferences is discussed. Moreover, in the thesis of Hu [16] voiced and unvoiced speech are separately processed.

Roweis [17] proposed a probabilistic approach for source separation. This system, know as factorial-max vector quantizer (FM-VQ), represents the log-magnitude spectrum of each source by a VQ. During separation the speaker dependent VQ models are combined into a factorial model to infer the most probable codewords observing the log-magnitude spectrum of the speech mixture. These codewords are employed to find a masking signal which is used for signal reconstruction. All systems using statistical models are known as model-based approaches. Since, source dependent models are employed for separation, model-based systems are mainly applied to co-channel speech. Existing model based approaches have the following shortcomings:

1) Due to the factorial nature, the computational complexity is a bottleneck and restricts its application in systems with real-time demands.

2) The best performance is achieved for equal signal mixing levels, due to the matched condition between training and testing.

3) The speaker identities must be known *a priori*, i.e. the systems are speaker dependent.

This thesis addresses the first two of the three mentioned shortcomings. Therefore, we first carry out an analysis of model-based approaches based on the FM-VQ model and address the problem of signal reconstruction. In this respect, we propose the soft binary mask for signal separation instead of the binary mask, which is suggested as computation goal for CASA by Hu et al. [18, 19]. We evaluate the binary, soft binary, and soft masks for reconstruction in terms of the target-to-masker ratio on artificial mixtures [20]. Additionally, we apply the masks to the *SAIL real life SCSS corpus* [21], which is a compilation of different television newscasts. We evaluate the separation quality by measuring the word error rate (WER) of the automatic speech recognition (ASR) system [22]. Compared to the WER of the real-world mixture and to the other masks, the soft binary mask shows superior performance. Furthermore, the model complexity of FM-VQ to represent each speakers characteristics is studied and experimentally evaluated. We place a main emphasis on the computational complexity in factorial models. This complexity arises from the observation likelihood computation during decoding. In order to reduce this computational complexity in factorial models we present two approximation techniques. They are either based on a modification of beam search or the iterated conditional modes algorithm to efficiently approximate the observation likelihood. Experimental results show only a marginal reduction in separation quality compared to the naive approach while the computational complexity is reduced by up to two orders of magnitude.

We propose the shape-gain model for source separation to tackle the problem of different mixing levels as well as the issue that same shapes at different gains are modeled separately. In this model the mixed signal is assumed to be a density function modelled by a random process with two latent variables. In order to separate the spectrum of each speaker, the gains are estimated for all component distribution. The spectrum for each speaker is finally obtained by the expected value of the constructed multinomial distribution observing the mixture distribution. For gain estimation various approaches are discussed and evaluated.

Finally, the incorporation of low-level cues of CASA in model-based methods is a key objective of this thesis. Therefore, the signal is decomposed into a source- and a filter-related part. The source or excitation signal represents the vibrations of the vocal folds, whereas the filter models the shaping of the vocal tract. To perform separation, sequential and parallel graphical model structures are employed. Additionally, the source-filter and the shape-gain representations are combined, which reflects the speech production model [23].

1.2 Related PhD Theses

This section discusses the scope of this work with respect to other selected theses in the field of single channel source separation. The selection is based on either closely related theses or theses with significant contributions for SCSS. The aim of this section is to give an overview to related work, to discuss similarities and to emphasize extensions and novel contributions to SCSS.

1.2.1 M.H. Radfar: Single Channel Speech Separation

The thesis of Radfar purely discusses the single channel source separation problem. This thesis is not written in English, therefore the author assured that the main matter of the thesis is summarized in two papers [24, 25]. The work can be mainly divided into two parts: The first part deals with source-filter based source separation. The source driven part is modeled by a multi-pitch tracking unit [26]. For this purpose, no probabilistic representation is employed. Specifically, a sinusoidal representation is extracted from the speech mixture. For every time frame, only a specific number of sinusoids are allowed. These sinusoidal candidates have been merged to form trajectories over time. The formation of trajectories is based on a heuristically specified frequency range. Specifically, a sinusoidal candidate at time $\tau + 1$ has to be within a specified frequency range of a candidate at time τ. In the second unit, a harmonic modeling strategy is followed in order to find pitch estimates. Therefore, a minimum mean square error approach for the joint estimation of two pitch values is applied. According to the author, the harmonic estimation suffers from the following: (i) pitch doubling/halving, (ii) masking of one speaker, and (iii) identical pitch or harmonically related pitch. For this reason, the output of the first and second unit is compared and only those sinusoidal trajectories remain, which share a minimum number of pitch candidates extracted by the second unit. Since this sequence of heuristics may lead to more than two pitch tracks, the author proposed to use a single pitch tracking algorithm, to restrict the number of pitch tracks to the two most promising ones. The following list of issues may raise concerns for the multi-pitch tracking unit:

- Evaluation: This method is evaluated only on a small dataset, which might be restrictive.

- Error measure: The method is compared to the method of Wu et al. [27]. Wu suggested an error measure for multi-pitch tracking, which comprises several submeasures. Radfar employed two of them to assess performance, namely, the gross error rate (E_{gross}) and fine error rate (E_{Fine}). Instead, the whole error measure of [27] should have been investigated for performance assessment.
- Speaker assignment: The proposed method can extract two simultaneous pitch tracks. However, an assignment of pitch tracks with unvoiced sections in between to specific speakers, seems to be impossible. Thus, we expect the method to be prone to permutation errors.
- Voicing decision: It seems that the voiced-unvoiced decision is made inherently by the sinusoidal representation unit. However, no results on this issue are discussed.
- There is no strategy given for cases where the pitch of both speakers is within the specified frequency range for pitch trajectory formation.

Once the two pitch tracks are estimated, they are employed for SCSS. To this end, excitation signals are synthesized. SCSS is carried out by including the excitation signals in the statistical model, which characterizes the vocal tract [28]. Recently, Radfar extended this model by a gain estimation unit [24]. In particular, a maximum-likelihood approach for gain estimation is followed. Therefore, the separation is basically carried out for a predefined set of mixing conditions and the mixing level maximizing the MAP estimate averaged over the complete utterance is selected. This seems quite restrictive because of the increased computational complexity and fixed mixing levels. Moreover, this method is not suitable for online speech separation.

In chapter 6, we basically follow this sequential source-separation approach, with significant extensions and modifications sketched in the following. First, for the source-driven part, double pitch tracking is carried out using a probabilistic model [29, 30]. This approach is compared to the method of Wu et al. [27] using the error measures proposed in their work [27]. Additionally, we propose a modification of this error measure which accounts for speaker assignment errors [31], called permutation error. Evaluation was performed for speaker dependent (SD), gender dependent (GD), and speaker independent (SI) trained models. For modeling the vocal tract related part, we also investigated a vector quantization (VQ) or a Gaussian mixture model (GMM) and, additionally, we study nonnegative matrix factorization (NMF). While for NMF the contribution of each basis also determines the gain factor for every time frame, for VQ the gain has to be additionally estimated. For this purpose, we propose an auditory motivated gain estimation strategy and a nonlinear based method in chapter 5, section 5.2.6 and 5.2.2. Both methods perform gain estimation on a segment level and not on the whole utterance.

The source separation methods in this thesis are compared in terms of the target-to-masker ratio (TMR) on the Grid Corpus [20]. Separation experiments have been conducted separately for the SD, GD and SI extracted pitch trajectories. Performance is also compared to the separation performance using the reference pitch values, extracted from the single speaker component utterances. This case is more or less the upper bound for pitch based source separation. Moreover, a tight relation between pitch estimation performance and separation performance is shown.

Presumeably the most striking difference is the used sampling frequency. While the system of Radfar is working at 8 kHz sampling frequency, the system discussed in this thesis is based on 16 kHz. Note, that the harmonics of a voiced speech signal are not resolved, i.e. not visible, above about 4 to 5 kHz. In a further extension of the source model, the excitation signals are directly represented by statistical models. This avoids multi-pitch estimation. For this system, we show that the sum of the excitation signals is a valid approximation of the spectrally whitened speech mixture. Note, the direct modeling of the source-driven part enables source separation also for unvoiced speech.

The second part of Radfars thesis was concerned with an optimal MMSE estimator, which is not relevant for this thesis.

1.2.2 M. Reyes-Gomez: Statistical Graphical Models for Scene Analysis, Source Separation and other Audio Applications

The thesis of Reyes-Gomez [32] is structured in three parts: (i) Multi-channel signal separation, (ii) single channel source separation based on the decomposition of the signal into subbands, and (iii) separation using local spectral deformations. Only the last two listed parts are relevant for this thesis and are shortly discussed.

Reyes-Gomez proposed an extension of the widely used factorial-max HMM model of Roweis [17]. In this extended model, the observation, i.e. the spectrum, is decomposed into bands. These bands are either treated independently or are synchronized, by modeling their dependency. The band representation not only increases the accuracy in modeling spectral details but also reduces the state space. For each subband, a separate HMM is trained during inference. In order to prevent unnatural subband combinations, Reyes-Gomez introduced a coupling between and within subbands to enforce consistency. This model is called coupled factorial HMM. Exact inference is intractable in the multiband model. Therefore, an approximation based on variational methods is developed [33], which results in a expectation-maximization like update scheme. Experiments with different numbers of subbands, show a superior performance only for the coupled model compared to the factorial-max HMM [17] approach.

In the last part of his work, an accurate description of source signals is given, avoiding the enormous size of dictionary prototypes (see also [34]) by exploiting the slow time variation of speech, i.e. the similarity of adjacent time frames. The slowly changing energy of speech across time and frequency is modeled by spectral deformations. To this end, a patch of neighboring previous time-frequency bins centered around the d^{th} frequency bin is employed to predict a patch of current time-frequency bins also centered around the d^{th} frequency bin. For the transformation, the current patch is assumed to be smaller than the previous. Thus, this model selects a transformation template from a discrete set, which best describes the evolution of the observed part of the spectrum. The whole system can be naturally represented as a graphical model. For every time-frequency bin a continuous variable is defined and discrete hidden transformation variables model the smooth deformations using Markov random fields. This model has been successfully applied for noise reduction. Moreover, the model has been generalized for missing data, where some continuous variables are allowed to be hidden.

Finally, a two-layer source-filter transformation model is introduced, where a separate two-layer deformation model is used for the source- and the vocal-tract related signal. For inference and tracking, loopy belief propagation is used. In an automatic speech recognition (ASR) experiment with noisy data, only a marginal reduction of the word error rate (WER) for different signal-to-noise-ratio (SNR) conditions, for both, the single-layer and two-layer model is achieved. Moreover, semi-supervised single channel source separation can be performed by manually selecting speaker dependent frequency bins [35].

1.2.3 S.J. Rennie: Graphical Models for Robust Speech Recognition in Adverse Environments

This PhD thesis is concerned with robust speech recognition in adverse environments [36] and mainly discusses this issue from the multi-channel perspective. However, the author and his colleagues presented the best performing system [5, 37] for the single channel speech separation challenge in 2006. This system exceeds human performance on this specified task. The main building blocks consist of an FM-GMM, which models the acoustic states and a grammar model on top of the acoustic model. The grammar dynamics are left-to-right phone models, which are trained using the word sequence defined by the *Speaker Separation Challenge* grammar [20]. For the mapping from words to three-state phone models, a finite state machine was employed. The usage of dynamics was investigated for either, the grammar or the acoustic model, or both. Moreover, for the acoustic model, an observation likelihood approximation was proposed, which is discussed and compared to the approximations of this thesis in chapter 4. To make the system applicable for the *Speaker Separation Challenge*, a gain estimation and speaker identification unit was integrated. The mixing level measured as SNR is estimated for the whole speech utterance *a priori*. Therefore, speech frames originating from a single source are identified. In this system, speaker identity and mixing level are estimated simultaneously on just these frames. This is accomplished in an expectation-maximization like fashion. In the expectation step the likelihood for a hypothesized speaker pair is calculated, whereas in the maximization step the appropriate mixing level is selected from a discrete set. Before separation the identified speaker dependent models are globally adjusted by the estimated SNR. Therefore, the whole utterance must be available in advance. Hence, this method can not be applied for online separation. In this work, the segmental estimation of the gain is proposed. The segmental estimation addresses both issues, the selection of the global mixing level from a discrete set and the application for online processing.

1.3 Organization

This thesis has six chapters and is structured according to the modules used to build up the final source-filter based separation system. After this introductory chapter, an overview of relevant work in single channel source separation is presented in chapter 2. Hereafter, model-based single channel source separation is reviewed and associated problems are discussed in chapter 3. Particularly, all three state-of-the-art interaction models are discussed, for signal reconstruction a masking signal is proposed and the problem of complexity in factorial models is introduced. Accordingly, chapter 4 deals with this issue, and adequate heuristics to approximate the observation likelihood will be introduced. The problem of different mixing levels emerges from these chapters. Therefore, chapter 5 discusses various approaches to estimate the gain for each speech segment from the speech mixture. Chapter 6 proposes the source-filter representation of speech signals, in order to separate the speech mixture. Therefore, sequential and parallel source as well as filter estimation strategies are used. Consequently, the observation likelihood approximations, as well as the gain estimation is integrated into the source-filter model. At the end of each chapter, the performance of the introduced methods is evaluated using the experimental setup described in section 1.5. Additional results for the source-filter algorithm applied to real-world recordings with various "speaker"-microphone distances are presented in Appendix A. Finally, Appendix B evaluates the developed SCSS algorithm in terms of the word error rate on the *SAIL real life SCSS corpus* [21] and on artificial speech mixtures defined on the WSJ0 corpus [38].

Parts of the thesis have been published previously:

- M. Stark, F. Pernkopf, T. V. Pham, G. Kubin. Vocal-Tract Modeling For Speaker Independent Single Channel Source Separation. In *IAPR Workshop on Cognitive Information Processing (CIP)*, pages 217-220, Santorini, Greece, June 2008.
- M. Stark and F. Pernkopf. Towards Source-Filter Based Single Sensor Speech Separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 97-100, Taipei, Taiwan, April 2009
- M. Stark and F. Pernkopf. On Optimizing the Computational Complexity For VQ-based Single Channel Source Separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 237-240, Dallas, Texas, April 2010
- M. Wohlmayr, M. Stark, F. Pernkopf. A Mixture Maximization Approach To Multipitch Tracking With Factorial Hidden Markov Models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5070-5073, Dallas, Texas, April 2010
- M. Stark, M. Wohlmayr and F. Pernkopf. Source-Filter Based Single Channel Speech Separation Using Pitch Information. In *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, accepted for publication
- M. Wohlmayr, M. Stark and F. Pernkopf. A Probabilistic Interaction Model for Multi-Pitch Tracking with Factorial Hidden Markov Models. In *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, submitted

1.4 Applications

In general, source separation is applied, whenever a certain source in a source mixture is of interest for further processing. The single channel separation of speech signals in particular has potentials in the field of speech coding, automatic speech recognition, and human hearing aids.

- **Speech Coding:** Speech coders are part of the family of source coders. In source coding, attributes of the source signals are used to remove redundancies and to represent the data efficiently. For example, the *adaptive multi-rate wideband speech transcoding* (AMR-WB) [39] speech coder utilizes a long-term and a short-term predictor for coding. This is related to the source-filter representation of speech signals. However, the coding fails, if the input of the coder is a mixture of two speech signals mixed at approximately equal level. To give an example, we consider a short time segment where both speakers utter voiced speech. In this case, the long-term predictor will code the fundamental frequency of the speaker with higher energy and the short-term predictor will encode the envelope of the combined speech signals. A proper source separation could offer a solution for this problem and provide only the speech of the target speaker as input.
- **Automatic Speech Recognition (ASR):** The automatic recognition of spoken words by machines has been a goal for a long time. The performance of modern ASR systems is acceptable for constrained tasks such as command and control applications. In such systems, only a limited size of vocabulary is used. For unconstrained tasks, the output of ASR systems is not that reliable [40, 41]. In acoustic environments with background

interferences, the speech recognition performance decreases dramatically. Even more, when the interference consists of single or multiple speech sources. To alleviate the impact of such interferences in ASR systems, a preprocessing unit in the front-end attempts to remove the interferences and to provide a clean signal as input to an ASR.

- **Human Hearing Aids:** Hearing impaired listeners wearing a hearing prosthesis suffer from sensitivity losses or experience extreme difficulties in auditory scenes with multiple active sources. Under these conditions, hearing impaired listeners need 4.2-10 dB better SNR for the same intelligibility than normal hearing individuals [42]. The lost sensitivity can be restored by amplification and dynamic range compression, while for the separation problem a beamforming approach is utilized which enhances voices from a particular direction [43]. The additional use of single channel source separation could alleviate the cocktail party problem [44] elegantly and pass an estimate of the target speech to the ear. Summarizing, SCSS could further improve speech quality in hearing aids, if applied additionally to the used beamforming technique.

While for hearing aids, source separation could be employed to improve speech intelligibility, for ASR, the recognition performance should be maximized. For noise reduction algorithms the difference between human and machine applications has been studied for example in [45, 46]. This difference is less studied for single channel source separation.

1.5 Experimental Setup

In all source separation experiments throughout this thesis, the Grid Corpus provided by Cooke et al. [20] for the SCSS task has been selected. This database contains separate sets for training, testing, and development. The Grid corpus consists of 34 talkers, each uttering 1000 sentences. Thus, the total corpus size is 34 000.

Most evaluation criteria compare the separated signal estimates to the reference or source signals. Additionally, we assess performance of the multi-pitch tracking unit using the true reference pitch tracks (see chapter 6, section 6.2.1). Since for the test data only the speech mixtures only are available, we use data from the training corpus for training and testing.

As preprocessing step, we resample the database from 25 kHz to 16 kHz. For this task, the MATLAB routine *resample* is used. For spectrogram calculation, the signal is cut into segments of 32 ms with time shifts of 10 ms. Afterwards, the speech segments are multiplied with a Hamming window and transformed to the frequency domain using the discrete Fourier transform [47]. We denote the complex spectrogram with upper case symbols, i.e. Y and S, the magnitude spectrogram with lower case bold symbols, i.e. $\mathbf{y}$ and $\mathbf{s}$, and the log-magnitude with upper case bold symbols, i.e. $\mathbf{Y}$ and $\mathbf{S}$.

Whenever speaker independent models are trained, we use 10 male (MA) and 10 female (FE) speakers, each producing at least 2 minutes of speech. The labels of the speakers, as specified in the data set [20], are shown in Table 1.1. Two randomly selected male and female

Table 1.1: Label of female and male speakers used for training speaker independent models.

	speaker									
FE	4	7	8	11	15	16	21	22	23	24
MA	3	5	6	9	10	12	13	14	17	19

Table 1.2: Labels of speakers and file names used for testing.

FE1	speaker 18	"lwixzs"	"sbil4a"	"prah4s"
FE2	speaker 20	"lwwy2a"	"sbil2a"	"prbu5p"
MA1	speaker 1	"pbbv6n"	"sbwozn"	"prwkzp"
MA2	speaker 2	"lwwm2a"	"sgai7p"	"priv3n"

speakers, each uttering 3 sentences as shown in Table 1.2 are used for testing. For simplicity, we will call these speakers FE1, FE2, MA1 and MA2 in the sequel.

The sentence structure is given as: "<command> <color> <preposition> <letter> <digit> <adverb>" [20]. For instance, the label "lwixzs" identifies the sentence "lay white in x zero soon". The spectrogram, orthographic transcription and time-domain signal of this sentence is shown in Figure 1.3. This is an extension to the corpus for coordinate response measure [48]. Statistical models capturing speaker dependent characteristics are trained using 6 min of speech material each. To determine a statistical representation of the data, either the EM-algorithm [49] or the k-means [50] algorithm can be used. This results either in a Gaussian Mixture Model (GMM) or a Vector Quantizer (VQ). Since the GMM models show lower separation performance, we employ VQ models in the experiments. For models with memory, i.e., first order Markov chains, the transition probabilities are obtained by counting and normalizing. Prior probabilities are determined in the same way. Moreover, we apply Laplace smoothing [51, 52], a method of discounting, i.e., probability mass is moved from observed to unobserved events. This method addresses the sparse data problem by adding the count of 1 to each observation.

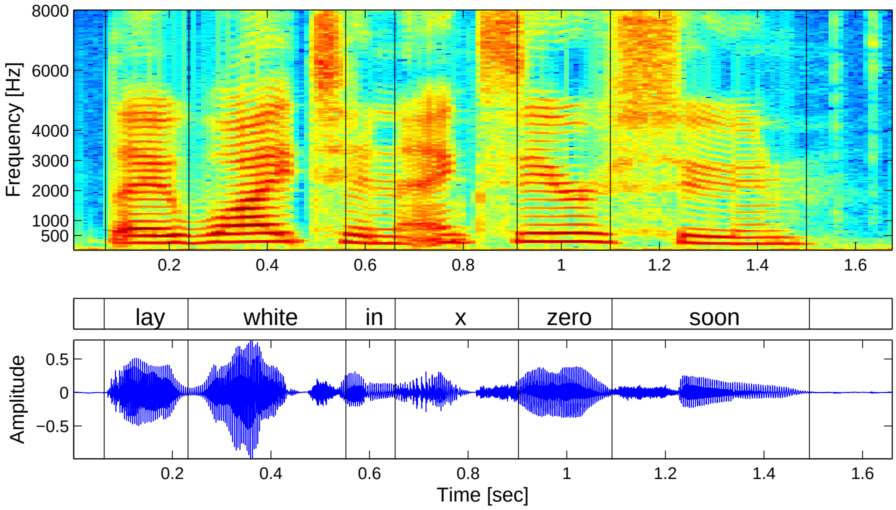


Figure 1.3: Spectrogram, orthographic transcription and time-domain signal of the sentence "lay white in x zero soon" of speaker FE1.

To evaluate the speech separation performance, the target-to-masker ratio (TMR) has been used. To avoid synthesis distortions affecting the quality assessment, the TMR has been measured by comparing the magnitude spectrograms of the true source and the separated signal as:

$$\mathrm{TMR}_i = \frac{\sum_{\tau,d} (\mathbf{s}_i^{(\tau,d)})^2}{\sum_{\tau,d} (\mathbf{s}_i^{(\tau,d)} - \hat{\mathbf{s}}_i^{(\tau,d)})^2}, \tag{1.2}$$

where $\mathbf{s}_i$ and $\hat{\mathbf{s}}_i$ are the source and separated signal spectra of the considered speaker i, and τ and d are the respective indices running over time and frequency. All possible combinations between target speakers and their interfering speakers are evaluated, resulting in altogether 54 speech mixtures for 4 speakers and 3 utterances each. Hence, 108 separated component signals are used for evaluation. For testing, all files are mixed at equal level of 0 dB TMR if not specified otherwise.

In all figures, the achieved mean value is depicted with a red horizontal line. The methods are identified by the label on the horizontal-axis. Moreover, the standard deviation of the TMR is indicated by the blue box surrounding the red line. All experiments are split into three classes:

- Same gender female (SGF)
- Same gender male (SGM)
- Different gender (DG)

An example result plot is shown in Figure 1.4.

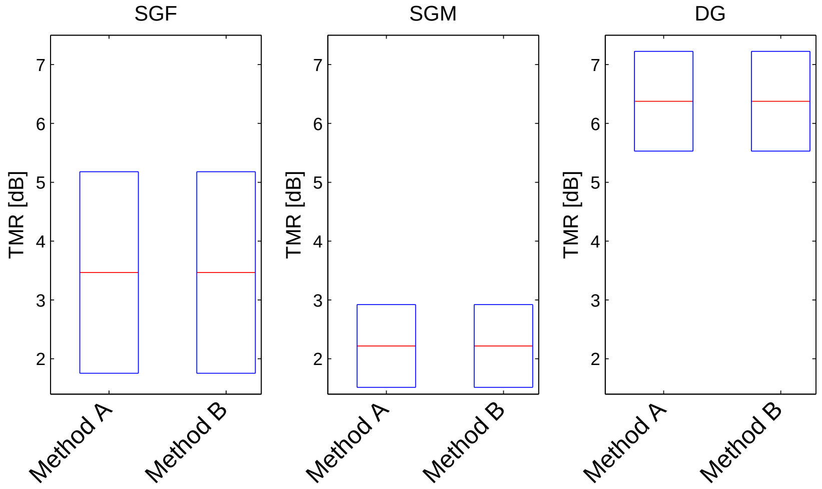


Figure 1.4: Example result plot. Same gender female (SGF), male (SGM), and different gender (DG) results are plotted separately. The labels identify different methods.

Chapter 2

An Overview of Work in Single-Channel Speech Separation

This chapter presents an overview of existing single-channel or monaural source separation methods. This overview is not restricted to methods for speech separation only but also covers music signal separation methods. The sequel is structured in four parts: The first part discusses methods which try to mimic the human auditory system. Therefore, sophisticated models are developed to integrate source specific signal characteristics. The second part discusses unsupervised learning methods for separation. Algorithms based on basis decomposition as well as probabilistic approaches are discussed. Afterwards, the modulation spectrum as a unitary signal transform will be discussed for single-channel source separation (SCSS). In the literature, different objective quality measures to assess performance for single- and multi-channel source separation have been introduced. Therefore, the last section discusses relevant objective and subjective quality measures.

2.1 Computational Auditory Scene Analysis (CASA)

The cocktail party problem (CPP) as a psychoacoustic phenomenon is related to the human ability of selective hearing. Humans are not only able to select a source of interest in an auditory scene but can also track and identify this source [7, 44]. In computational auditory scene analysis (CASA) [3] the attempt is made to computationally model the human auditory system and its processing in the brain. The goal in CASA is to incorporate as much information as the human auditory system is using. This starts with the selection of appropriate features like onset, offset, fundamental frequency, amplitude and frequency modulation, position, continuity and harmonicity. However, for the SCSS task no spatial information can be used. Bregman [8] first systematically explains the perception of complex acoustic mixtures.

As stated in [44], CASA systems can be mainly split into data-driven and prediction-driven systems. Both are exploiting low-level acoustic cues for grouping. Grouping refers to the task of finding speaker unique cliques. Grouping can be divided into, (i) simultaneous grouping, which aims at finding speaker unique frequency cells for each time step and (ii) sequential grouping, which denotes the process of forming speaker unique streams. The main difference between data- and prediction-driven systems is that data-driven systems are extracting time-frequency patterns of the auditory scene and are then performing the grouping whereas in prediction-driven approaches, prediction is based on a world model representing the stimulus.

CASA approaches suffer from two problems in speech separation: Firstly, the separability is rather limited for unvoiced speech and secondly the formant structure is not explicitly used as a feature [44]. In the next sections two applications of CASA methods are introduced.

2.1.1 Temporal Binding and Oscillatory Correlation

Van der Malsburg introduced the theory of temporal binding in 1981 [53]. In this theory he suggested that the binding mechanism between presynaptic and postsynaptic activities can be explained by correlations [44]. The strength of synapses is time dependent, i.e. it is increasing with time, and thus follows the Hebbian postulate of learning [1]. Further, a temporal synchrony between presynaptic and postsynaptic neurons is assumed, called *Malsburg Synapse*. This is taken as a basis to form a topological network where the synchronization mechanism allows the neurons to be linked in multiple active groups simultaneously. The temporal binding problem is suggested to be solved by a linking architecture letting neurons fluctuate temporarily and binding together synchronized neurons into higher-level symbols.

The idea of oscillatory correlation was first proposed by Van der Malsburg. In the CASA application, different sensory domains are used for binding of sensory components. A two-layered oscillator network was introduced by Wang and Brown [12] in 1999 for this task. Their method performs segment formation and stream segregation based on oscillatory correlation. In the oscillatory correlation-based model a stream is represented by a population of synchronized relaxation oscillators, which are corresponding to auditory features. Different streams are represented by desynchronized oscillatory populations.

The method is also compared to BSS techniques and exhibits comparable performance [54]. Two drawbacks of this method are the performance drop in multiple competitive environments and a separation performance depending on the source signal itself. In summary, under most noise conditions the tested BSS techniques outperformed the oscillatory model as long as the ICA assumptions, of the BSS techniques are not violated.

2.1.2 Cortronic Network

The cortronic network [44, 55] consists of an artificial neural network, which is motivated by the sparse coding scheme the human brain employs [56] when extracting features of sensory inputs and accessing them through associative memory. Such a biologically motivated model has been recently developed by Sagi [55] to solve the CPP by machine. It is represented by an associative memory neural network model consisting mainly of three distinct layers: (i) sound-input representation region, (ii) sound processing region and, (iii) word processing region. This method rests on two assumptions: Firstly, the network needs knowledge about the speech signals (i.e. the language the recognizer should understand/listen to) and secondly, the methodology used to design the network is based on the framework of associative memory and pattern identification. To solve the CPP three different kinds of features are used: (i) sound and subsequent sound, (ii) sequence of sounds and, (iii) certain word and the following word in the language. These features are calculated using a wavelet-like transform to convert the sound input into activations used in the cortronic network. Finally, the task of the system is to focus on a particular speaker and isolate the word series which is uttered. For a particular instant of time, the expectation of each speaker's sound character [2] to be a part of

[1] This postulate discusses the relation between some kind of associative learning and simultaneous activation of cells, which leads to an increased synaptic strength.

[2] A sound character refers to a single or multiple phonemes.

the sound stream of the target speaker has to be computed. It is reported that this method is quite robust in terms of speaker, speech, and noise variations, even for signal-to-noise ratios (SNRs) under -8 dB. The SNR is introduced in section 2.4.1.

Bio-Inspired Sound Source Separation: The collaborative work [57] between University of Sherbrooke and Graz University of Technology is quite similar to the Cortronic Network from the motivation point of view. Therefore, it is included in this section. In contrast to other CASA methods, a finite impulse response (FIR) instead of an infinite impulse response (IIR) Gammatone Filterbank is used as feature extraction. This representation is said to result in less distortions for the reconstructed or synthesized signal, due to phase corrections. Furthermore, in this approach more filterbank channels than usually are used. As features, no specific CASA cues like onset, offset, or pitch are used, instead two maps namely either the Cochleotopic/AMtopic or the Cochleotopic/Spectrotopic map depending on the input sound object are used as input of the spiking neural network. The output of the neural network is a mask applied in the reconstruction phase to divide the signals. For this method no performance measure is available.

2.2 SCSS based on Basis Decomposition

A general problem in statistics and signal processing is to find a suitable and appropriate transformation of the data. For an observed random vector $\boldsymbol{S} = (s_1, s_2, \ldots, s_N)^T$ of N-sources and its linear transform $\boldsymbol{Y} = (y_1, y_2, \ldots, y_M)^T$ the goal is to find the weight matrix $\boldsymbol{W}$ in order to decompose the mixture, where M specifies the number of observations or sensors. The task of source separation is to recover the N-sources form M observed sensor signals. We assume the following linear relation between the observations and the source signals:

$$\boldsymbol{Y} = \boldsymbol{W}\boldsymbol{S}. \tag{2.1}$$

Several methods have been developed to find such a representation for different objectives. Principle component analysis (PCA) [58], as the correlation based transform, aims to decorrelate the input signals in $\boldsymbol{Y}$, which can be further used for dimension reduction. However, the application of such methods is in general restricted to applications, where the number of observations in $\boldsymbol{Y}$ is equal or greater than the number of unknown source signals. PCA, Non-negative matrix factorization (NMF), and independent component analysis (ICA) have been successfully applied for signal separation. However, these methods can be partially adapted to underdetermined systems such as SCSS. In the next section, ICA is introduced, which intends to make the transformed signal components statistically as independent as possible. Thus, ICA recovers statistically independent source signals. However, to enable separation, most of the SCSS methods additionally include source dependent prior knowledge learned during an enrollment phase.

2.2.1 Independent Component Analysis

The Independent Component Analysis (ICA) algorithms are data driven methods and perform best when the number of observed signals is equal or greater than the number of sources [4, 59]. ICA relies on the assumption of mutual statistical independence of the signals to separate. Based on this assumption, the goal is to find a transformation separating the mixed signals in a way that they are as much statistically independent as possible. Thus, ICA is a special case of redundancy reduction and provides a tool to estimate the unmixing matrix $\mathbf{A} \approx \mathbf{W}^{-1}$, assuming $\mathbf{W}$ is invertible. An introduction of ICA can be found in [60]. ICA cannot be directly applied for SCSS, where the number of observations is $M = 1$ and the number of sources is usually $N \geq 2$.

To overcome the limitations of under-determined systems, in general source *prior* knowledge is incorporated. In [17] a "refiltering" technique has been introduced to estimate a time-varying mask, which is used to separate the sources. In the following, methods either based on time domain or on frequency domain basis functions are considered.

Independent Subspace Analysis

Independent Subspace Analysis (ISA) was originally proposed by [61] for the application on images. ISA combines ICA and an invariant feature extraction. Casey [62] introduced ISA to single-channel source separation and extended it. The first extension, is the extraction of statistically independent subspaces from the projection of a one-dimensional signal onto a manifold. The second extension is the use of dynamic independent components to represent non-stationary signals. The dynamics are gathered by tracking the similarities of dynamic components over small time steps.

In this method, the instantaneous mixture signal is transformed to the frequency domain using the short-time Fourier transform (STFT). Afterwards, independent basis vectors are determined using ICA. The basis vectors $\boldsymbol{z}_i$ are assumed to be static but are weighted by a time-varying factor $\beta_i^{(\delta\tau)}$ as:

$$\mathbf{y}^{(\tau)} = \sum_{i=1}^{\rho} \beta_i^{(\tau)} \, \boldsymbol{z}_i, \tag{2.2}$$

where $i = [1, \dots \rho]$ denotes the index of basis vectors, ρ is the number of basis vectors, $\mathbf{y}$ is the STFT transformed observed signal vector, and τ is the frame index. This method for calculating basis vectors can be extended to dynamic or non-stationary information. Therefore, it is assumed that each transformed signal frame is stationary and a block of l such frames is used for subspace decomposition by rewriting Eq. (2.2):

$$\mathbf{y}^{(\delta\tau, l)} = \sum_{i=1}^{\rho} \beta_i^{(\delta\tau, l)} \, \boldsymbol{z}_i^{(l)}, \tag{2.3}$$

where $\delta\tau$ denotes the block hop size, usually set to the half of the block length.

The appropriate number of basis vectors ρ is found by singular value decomposition and applying a threshold on the decreasing sorted eigenvalues. Finally, independent feature vectors are assigned to sources based on a similarity measure. The similarity is represented in an ixigram, which measures the mutual similarity of components in an audio segment as independent cross-entropy matrix. The pair-wise similarity measure is approximated by the symmetric Kullback-Leibler distance, resulting in a symmetric distance matrix D. Grouping is performed by a clustering procedure using the dissimilarities in the D matrix. The source signals can be reconstructed using the weights and the source dependent basis vectors.

Time Domain Maximum Likelihood ICA Approach

In [59, 63], the time domain ICA basis filters of the source signals are learned *a priori* from a training dataset. The data set consists of source dependent material only. The learned source basis filters are finally used to separate the mixed test utterances. Here, the assumption of two generative models is made: First, it is assumed that the mixed signal consists of a weighted sum of the source signals for every time instant. Secondly, it is assumed that this process can be reversed by:

$$y(t) = \sum_{t_f=t}^{T_f} \boldsymbol{w}_{t_f} s_{t_f}(t) = \boldsymbol{W} s(t), \tag{2.4}$$

where $s(t)$ is the independent assumed unknown source vector and $\boldsymbol{w_i}$ are called the basis functions generating the segment of the observed signal.t_f is the time index within a frame of T_f samples.

$\boldsymbol{A} = \boldsymbol{W}^{-1}$ refers to the ICA filters, where $\boldsymbol{W}$ is defined in Eq. (2.1). The ICA filters transform the segments into source coefficients $s(t) = \boldsymbol{A}\ y(t)$.

The goal of ICA learning is the maximization of the source specific densities $p(s_i(t))$ for given training data. Moreover, the source specific models are transformed such that they are statistically as independent as possible. Independency splits the joint probability in the product of marginals, i.e. $p(s_1(t), s_2(t)) = \prod_i p(s_i(t))$. Both, the information maximization principle [64] or the maximum likelihood estimation can be used for learning. In [63], the infomax rule with natural gradient extension for learning is used. To learn basis functions the time domain signal is cut into uniform length segments. Finally, the learned basis functions $\boldsymbol{A}_i$ can be employed for source separation. Using Bayes theorem the maximum a posteriori estimate can be found as:

$$\{s_1^*(t), s_1^*(t)\}_{t=1,\ldots,T} = \underset{s_1(t), s_2(t)}{\operatorname{argmax}}\ p(s_1(1), \ldots, s_1(T)|\boldsymbol{A}_1)\ p(s_2(1), \ldots, s_2(T)|\boldsymbol{A}_2), \tag{2.5}$$

where the most likely bases are estimated given the basis models $\boldsymbol{A}_i$. For separation the mixture is assumed to be a weighted sum of the component sources $y(t) = \lambda_1 s_1(t) + \lambda_2 s_2(t)$, where λ_i is the weight associated to each source. The following artificial mixtures have been used for performance evaluation: Rock-Jazz, Rock-Male, Rock-Female, Jazz-Male, Jazz-Female, Male-Female. The signal-to-noise ratio (SNR), as introduced in section 2.4.1 is employed for performance evaluation. For the Male-Female mixture a SNR improvement of 5.9 dB is reported. However, for the other mixtures, the SNR could be increased by up to 13 dB.

2.2.2 Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) is another method capable of identifying components with temporal structure. Among other methods, positive matrix factorization was first introduced in [65, 66] whose author also proposed an efficient algorithm for decomposition. The NMF algorithm decomposes a non-negative matrix $\boldsymbol{X} \in \mathbb{R}^{\geq 0, D\times T}$into two non-negative matrices $\boldsymbol{W} \in \mathbb{R}^{\geq 0, D\times R}$ and $\boldsymbol{H} \in \mathbb{R}^{\geq 0, R\times T}$, where $R \leq T$, such that a cost function

$$D(\boldsymbol{X}||\boldsymbol{W}\boldsymbol{H}) = \left\| \boldsymbol{X} \otimes \ln(\frac{\boldsymbol{X}}{\boldsymbol{W}\boldsymbol{H}}) - \boldsymbol{X} + \boldsymbol{W}\boldsymbol{H} \right\|_F \tag{2.6}$$

is minimized [67]. Here, $\boldsymbol{X}$ is the magnitude spectrogram of the signal, with D frequency bins and T time frames, $\otimes$ and $\overset{\cdots}{\underset{\cdots}{-}}$ are the element-wise product and division, respectively, $\|\cdot\|_F$ denotes the Frobenius norm, and R is a parameter describing the rank of basis decomposition. Perfect reconstruction is obtained if and only if $\boldsymbol{X} = \boldsymbol{W}\boldsymbol{H}$. This cost function equals the Kullback-Leibler divergence if and only if $\sum_{d,\tau} \boldsymbol{X}^{(d,\tau)} = 1$ and $\sum_{d,\tau} \boldsymbol{W}\boldsymbol{H} = 1$. Another appropriate cost function is the Euclidean distance.

Using this decomposition, one can observe that $\boldsymbol{W}$ is describing the vertical structure and $\boldsymbol{H}$ the horizontal structure of $\boldsymbol{X}$, i.e. the spectral and the temporal evolution of the parts of objects are modeled separately in the magnitude spectrum. $\boldsymbol{H}$ is a time-varying weighting of the extracted objects in $\boldsymbol{W}$. The update algorithm to optimize $\boldsymbol{W}$ and $\boldsymbol{H}$ is given by

$$\boldsymbol{H} = \boldsymbol{H} \otimes \frac{\boldsymbol{W}^T \frac{\boldsymbol{X}}{\boldsymbol{WH}}}{\boldsymbol{W}^T \, \boldsymbol{1}}, \qquad \boldsymbol{W} = \boldsymbol{W} \otimes \frac{\frac{\boldsymbol{X}}{\boldsymbol{WH}} \boldsymbol{H}^T}{\boldsymbol{1} \, \boldsymbol{H}^T}, \tag{2.7}$$

where $\boldsymbol{1}$ is the $D \times T$ matrix containing only unity entries.

Smaragdis [67] showed that this method is suitable for sound object extraction and can be further used for music transcription. In [68] performance evaluations of NMF for speech recognition have been carried out and compared to the factorial max vector quantizer (FM-VQ) method [69]. The authors report a superior performance of the FM-VQ approach compared to NMF measured in terms of the word error rate.

Convolutional NMF

To be able to describe also the temporal evolution of information, an extension to NMF was first introduced in [70] for sound object extraction and proposed for speaker separation in [71]. The reconstruction of $\boldsymbol{X}$ by the matrix product $\boldsymbol{X} \approx \boldsymbol{W} \cdot \boldsymbol{H}$ is extended to the convolutive NMF as

$$\boldsymbol{X} \approx \sum_{k=0}^{K-1} \boldsymbol{W}(k) \cdot \overset{k\rightarrow}{\boldsymbol{H}}, \tag{2.8}$$

where $\overset{k\rightarrow}{\cdot}$ is the right shift operator, which shifts the columns of a matrix to the right by k positions, introducing k columns of zeros at the left, and dropping k columns at the right.

For minimizing the reconstruction error the existing framework of NMF can be used and defined as:

$$D(\boldsymbol{X}||\hat{\boldsymbol{X}}) = \left\| \boldsymbol{X} \otimes \ln(\frac{\boldsymbol{X}}{\hat{\boldsymbol{X}}}) - \boldsymbol{X} + \hat{\boldsymbol{X}} \right\|_F, \tag{2.9}$$

where $\hat{\boldsymbol{X}}$ is the approximation of $\boldsymbol{X}$, defined as $\hat{\boldsymbol{X}} = \sum_{k=0}^{K-1} \boldsymbol{W}(k) \cdot \overset{k\rightarrow}{\boldsymbol{H}}$. The columns of $\boldsymbol{W}(k)$ are called bases.

Using the framework of NMF, speaker dependent bases can be derived from the magnitude spectrogram in the training phase. All bases trained by the speakers are finally merged to one matrix $\boldsymbol{W}(k)$. For a given mixture containing speech from known speakers but for unknown utterances, the goal is to find the weights $\boldsymbol{H}$. Thus, during separation the bases are fixed. Finally, the determined weights and the speaker dependent bases are used to reconstruct the magnitude spectrogram of each individual speaker as

$$\boldsymbol{Z}_i = \sum_{k=0}^{K-1} \boldsymbol{W}_i(k) \cdot \overset{k\rightarrow}{\boldsymbol{H}_i} \tag{2.10}$$

The phase of the mixture is used to compute the complex spectrogram of each reconstructed speaker. Finally, the inverse STFT is used to compute the time domain signals.

Several experiments have been performed using different numbers of bases R, different lengths of bases K, and different number of FFT bins D, to evaluate the performance of NMF. Unfortunately, no other research group uses the performance evaluation criteria introduced in [71], also described in section. 2.4.1, and no comparisons to other methods have been carried out.

2.2.3 Latent Variable Decomposition

Latent Variable Decomposition attempts to reconstruct entire spectrograms of each speaker from the observed speech mixture using source dependent prior knowledge. As prior knowledge, characteristic spectro-temporal structures are learned for each individual speaker [72] during a training phase. In the operating stage mixed signals are decomposed into a linear combination of the basis.

The speakers' spectral structure is identified through latent variable decomposition. This method is based on a statistical model where the spectral speech vectors are assumed to be the outcome of a discrete random process generating frequency bin indices. Therefore, the magnitude spectrum represents a scaled histogram of the number of draws of random variables associated to each speaker. The distribution of the random process is modeled as a mixture multinomial distribution, with time-varying mixture weights from frame to frame. To incorporate *a priori* knowledge in the training stage, the mixture multinomials are learned via the EM algorithm for each speaker separately. Hence, this method is referred to be a supervised method. To separate a mixed signal, maximum likelihood estimates of the mixture weights of multinomials and the *a priori* probabilities for each speaker and frame are obtained. The final spectrum for the speaker within a frame is obtained as the expected value of the number of draws of each frequency index from the speaker multinomial distribution.

For evaluation, 20 female and 20 male speaker models have been trained, each using 30 seconds of speech. This method achieved an improvement in SNR^f gain of 5.3 dB on average. The SNR^f used for this method is a frequency based signal-to-noise ratio and is defined later in section 2.4.1. Compared to the FM-VQ method in [69] which uses 30 times more training data and takes much longer to perform separation, the proposed method is said to outperform the FM-VQ approach perceptually. In [73] this method has also been used for singing voice and music separation. The goal of this work was not to measure separability but to further process the extracted singing voice (i.e. transpose the voice or change the gender of the voice).

2.3 Separation based on Modulation Frequency Analysis

The use of the modulation frequency analysis as representation of speech signals is motivated by psychoacoustic research [74], which claims that the human auditory system not only analyses but also separates signals in this domain. Schimmel et al. [75] present a source separation method to separate speech signals in the modulation frequency domain. The modulation spectrum is calculated by a filterbank, which is followed by a subband envelope detection unit and a frequency analysis of the subband envelopes. In the modulation frequency domain, a signal frame is represented in an acoustic frequency and modulation frequency plane. In this representation, two voiced speech frames separate well and a mask can be applied for each speaker to perform separation. Schimmel et al. [75] report good separability for voiced but restricted capability to separate unvoiced speech. To enable automatic separation, the frequency range of the fundamental frequency of the target and interfering speaker is assumed to be *a priori* known. Moreover, it is assumed that the frequency ranges should be sufficiently non-overlapping, which is rarely the case in a usual conversation scenario. According to the author, the subjective separation performance of speech signals mixed at equal level is high for voiced speech but is low during unvoiced parts such as transients.

2.4 Evaluation Methods

This section introduces quality assessment methods for single-channel speech separation. Quality measures are important to judge the separation quality of developed algorithms. In order to make algorithms of different research groups comparable it is of major importance to specify evaluation criteria and to make test databases publicly available. In general, one can distinguish between objective and subjective speech quality assessment methods. Subjective measures are found by listening tests. Therefore, a large number of listeners are required for reliable results. Listening tests are time consuming and expensive and, therefore, they are seldomly used. Objective quality measures can be divided into low-level measures and automatic recognition measures. Low-level measures can be further divided into intrusive and non-intrusive quality measures. In this section, we only deal with intrusive quality measures, where the underlying reference signals are available for comparison. Thus, it is not the goal to estimate quality measures from the mixture signals exclusively, i.e. from signals recorded under real environmental conditions with unknown source signals and mixture. Rather, we aim to introduce quality measures used and established in literature for artificial linear instantaneous mixtures.

First, low-level measures are reviewed in section 2.4.1. Afterwards, automatic recognition measures are presented followed by subjective quality assessment methods.

2.4.1 Objective Quality Measures

The signal-to-noise ratio (SNR) in the time domain

The first introduced performance evaluation method is the signal-to-noise ratio [76] between the separated and original signals in the time domain. The SNR is defined as:

$$\mathrm{SNR}_i = 10\log_{10}\left(\frac{\sum_{t=1}^{T}(s_i(t))^2}{\sum_{t=1}^{T}(s_i(t)-\hat{s}_i(t))^2}\right), \tag{2.11}$$

where i is the source index, T is the length of the speech signal, s and $\hat{s}$ denote the original and separated signal, respectively.

The signal-to-noise ratio in the frequency domain

The signal-to-noise ratio between the separated and original signals can be also defined in the frequency domain SNR^f [68], which is defined as:

$$\mathrm{SNR}_i^f = 10\log_{10}\left(\frac{\sum_{(\tau,d)} \mathbf{s}^{(\tau,d)}(i)^2}{\sum_{\tau,d}\left|\mathbf{s}^{(\tau,d)}(i)\ e^{j\psi^{(\tau,d)}} - \hat{\mathbf{s}}^{(\tau,d)}(i)\ e^{j\psi^{(\tau,d)}}\right|^2}\right), \tag{2.12}$$

where i denotes the source index, $e^{j\psi^{(\tau,d)}}$ is the phase of the mixed signal spectrum at time τ, and $\mathbf{s}$ and $\hat{\mathbf{s}}$ are the magnitude spectrum of the original and separated signal, respectively.

Segmental signal-to-noise ratio (SNR_{seq})

The global SNR measures discussed in the previous sections do not match well with human perception [77, 78]. The definition of the SNR on short-time segments is known as segmental

SNR (SNR_{seq}). This simple objective measure has proven to match better with human perception [77]. In the time domain, the SNR_{seq} is defined as

$$SNR_{seq} = \frac{1}{T}\sum_{\tau=1}^{T} 10\log_{10}\frac{\sum_{n=1}^{N} s^{(\tau)}(n)^2}{\sum_{n=1}^{N}(s^{(\tau)}(n) - \hat{s}^{(\tau)}(n))^2}, \tag{2.13}$$

where τ is the segment index and n denotes the sample index within a time segment.

Percentage of crosstalk suppression

The percentage of crosstalk suppression P_i is proposed in [76] to measure the separation quality. This measure quantifies the degree of crosstalk (interference) suppression between the two separated signals. Here, the binary masks for both the reference signals and the separated signals are computed. For the reference signals the mask is known as ideal binary mask. The binary mask is in detail discussed in section 3.4. Both binary masks are applied on the mixed signal. The signal, reconstructed using the ideal binary mask is denoted as s_i^{iBM} and the signal reconstructed with the estimated binary mask with s_i^{BM}. Then the crosstalk-to-signal ratio (CTS) is computed in the time domain as:

$$\text{CTS}_i = \frac{\sum_{t=1}^{T}(s_i^{\text{BM}}(t))^2}{\sum_{t=1}^{T}(s_i^{\text{iBM}}(t))^2}, \tag{2.14}$$

where $i \in \{1, 2\}$ is the speaker index. The percentage of crosstalk suppression can be computed as: $P_i = 100\,(1 - \text{CTS}_i)$.

Evaluation methodology by Smaragdis

Smaragdis [71] proposed three measures to judge the overall separation performance of SCSS algorithms. These are the speaker energy ratio, the similarity index and the residual energy. While the first measure tells us how much energy of the interfering speaker has been suppressed, the similarity index measures the similarity between reference and estimated signal. The residual energy measures how much energy has not been assigned to any speaker. Those measures are summarized in the following:

Speaker energy ratio: The speaker energy ratio is computed by the correlation between extracted signal $\hat{s}_i$ and the original signal s_j:

$$c_{i,j} = \text{cor}(\hat{s}_i(t), s_j(t)), \tag{2.15}$$

where $i, j \in \{1, 2\}$ are the source indices and cor denotes correlation. This correlation measure is used to define the speaker energy ratio SR as the logarithmic ratio of the correlation between extracted signal $\hat{s}_i$ and corresponding original signal s_i, and the sum of correlations to all other original signals $s_j, j \neq i$ from the target speaker:

$$\text{SR}_i = 10\log_{10}\frac{c_{i,i}}{\sum_{\forall j\neq i} c_{i,j}}. \tag{2.16}$$

A larger SR value indicates better extraction of the target source/speaker.

Similarity index: The second measure defined in [71] is the similarity index (SI), which describes how much the estimated output resembles the desired output. It is defined as:

$$\mathrm{SI}_i = 10\log_{10}\mathrm{cor}(\hat{s}_i(t), s_i(t)), \tag{2.17}$$

where SI always is equal or less than 0, with 1 being the most desired value.

Residual energy: To measure the variance of the difference between the input signal (speech mixture) and the sum of the extracted signals the residual energy (RE) has been introduced as:

$$\mathrm{RE} = \mathrm{var}(\sum_i s_i(t) - \sum_i \hat{s}_i(t)). \tag{2.18}$$

The RE measures how much of the extracted signals is accountable by the mixture signals. Values closer to zero indicate good accountability.

Estimated source decomposition for performance evaluation

In 2003 Vincent et. al [79] first proposed performance measurement methods for blind audio source separation (BASS), for both, single and multi-channel algorithms. In 2006 the same group [80] refined the definition of their methods as well as they introduced procedures to numerically calculate the quantities. In order to get a good representation of errors which may occur in BASS, several measures are proposed, each investigating a certain property of the error, e.g. distortion or interference energy.

Intersymbol interference (ISI) [81] a method established in BASS is said to be limiting because it is not applicable for underdetermined BASS. Moreover ISI is restricted to time-invariant linear mixtures. An L_2-norm based measure [82, 83] directly compares the time-domain source signal and its separated estimate, with focus on the indeterminacy of the task. This measure produces almost the same value for a broad range of separation qualities. Hence, poor results are evaluated rather coarsely. In order to have a measure independent on the number of channels and sources, new methods have been proposed by Vincent et. al [80]. The estimated source $\hat{s}$ is divided as follows:

$$\hat{s}(t) = s_{\mathrm{target}}(t) + e_{\mathrm{interf}}(t) + e_{\mathrm{noise}}(t) + e_{\mathrm{artif}}(t), \tag{2.19}$$

where $s_{\mathrm{target}}(t)$ is the target source, and $e_{\mathrm{interf}}(t)$, $e_{\mathrm{noise}}(t)$, $e_{\mathrm{artif}}(t)$ are the interference, noise and artifact errors, respectively. The decomposition of the estimated source signal is based on orthogonal projections. However, all these measures are source dependent and therefore have to be computed for each source separately. A procedure to calculate the pure source specific energy contained in the separated source signal is described in [80]. In the following, performance criteria in decibels are introduced.

Source-to-distortion ratio (SDR): The SDR measures the ratio of the target energy to all unwanted distortions comprised in the signal

$$\mathrm{SDR} = 10\log_{10}\frac{\sum_{t=1}^{T}\|s_{\mathrm{target}}(t)\|^2}{\sum_{t=1}^{T}\|e_{\mathrm{interf}}(t) + e_{\mathrm{noise}}(t) + e_{\mathrm{artif}}(t)\|^2}. \tag{2.20}$$

Source-to-interference ratio (SIR): The SIR measures the ratio between the target source component to all other source components in the mixture. In other words, the residual energy of one source given all others is computed as

$$\mathrm{SIR} = 10\log_{10}\frac{\sum_{t=1}^{T}\|s_{\mathrm{target}}(t)\|^2}{\sum_{t=1}^{T}\|e_{\mathrm{interf}}(t)\|^2}. \tag{2.21}$$

Sources-to-noise ratio (SNR):

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_{t=1}^{T}\|s_{\mathrm{target}}(t)+e_{\mathrm{interf}}(t)\|^2}{\sum_{t=1}^{T}\|e_{\mathrm{noise}}(t)\|^2} \tag{2.22}$$

Source-to-artifact ratio (SAR): Finally, the SAR is a measure to estimate the amount of distortions defined as

$$\mathrm{SAR} = 10\log_{10}\frac{\sum_{t=1}^{T}\|s_{\mathrm{target}}(t)+e_{\mathrm{interf}}(t)+e_{\mathrm{noise}}(t)\|^2}{\sum_{t=1}^{T}\|e_{\mathrm{artif}}(t)\|^2}. \tag{2.23}$$

Basically these measures are inspired by the SNR, but attempt to focus an specific aspects. One benefit of these measures is their independency. Similar definitions of these measures are given in [84].

To track performance changes within the unmixed audio signal, the measures can be used to numerically compute local performance measures on a frame bases. This may be beneficial in analyzing low performance regions.

Automatic recognition measures

If a particular application is already specified for the source separation algorithm, the performance is naturally judged by this application. In automatic speech recognition (ASR), the goal is the correct recognition of spoken words. The performance of ASR systems is usually measured by the word error rate (WER) defined as the percentage of the number of incorrectly recognized words over the total number of spoken words. The goal in ASR experiments is the minimization of the WER, in contrast to the target-to-masker ratio as defined in chapter 1, section 1.5, where the distortion energy should be minimized [85]. However, as the distortion energy approaches zero also the WER is minimal and the solutions are identical. The monaural speech separation and recognition challenge [37] was operated to perform a comparison of SCSS algorithms for the competing talker problem. For this challenge the Grid Corpus provided by Cooke et al. [20] was selected as database and an ASR system trained on isolated words was provided to assess performance. In general, the input feature vector of an ASR system reflects the movements of the vocal tract during articulation. Therefore, the mel-frequency cepstral coefficients or the linear prediction cepstral coefficients are used as features. Virtanen [86] and Deoras et al. [87] directly estimated the feature input vector of the ASR system from the speech mixture.

2.4.2 Quality Evaluation by Subjective Tests

To assess the quality of an SCSS method or the performance of a CASA method, subjective listening tests by humans are used. The goal is to compare the intelligibility between the separated signal and the mixed signal. However humans can not “turn off” the ASA process and hence subjective listening tests have to be carried out carefully. One possibility could be to use listening impaired and normal hearing humans [3] to measure speech intelligibility. Another could be the test of normal hearing individuals in very noisy environments such that they hardly can understand the spoken words. Ellis [88] asked listeners to score the resemblance between the segregated sound and the corresponding original sound part of a mixture. Two other methods to subjectively measure the speech intelligibility are the speech perception in noise test (SPIN) [89], which evaluates the word recognition in context. Therefore, 25 sentences with high and low context each are presented to the listeners with the task to repeat the last words of the sentence which is mixed with multi-speaker babble. The second example is the hearing in noise test (HINT) corpus [90], consisting of 25 phonetically balanced lists, each comprising 10 sentences mixed with speech-shaped noise.

Subjective listening tests are also carried out to get an understanding of the overall subjective quality of separated signals. Therefore, the mean opinion score (MOS) is in general used [91]. These tests have been mainly developed for speech coding purposes but can also by applied for noise suppression [92, 46] and source separation methods. Listening opinion tests, like the absolute category rating, comparison category rating, or the degradation comparison rating are recommended to get an overall impression of the speech quality.

2.5 Conclusion

This chapter aimed at providing an overview of the state-of-the-art in single-channel source separation. After we pointed out the main difference to multi-channel approaches which can additionally incorporate spatial information as cue, CASA based and model-based methods have been introduced. In addition to the performance of each method, benefits and drawback have been emphasized, such as the prior knowledge the methods rely on, e.g. speaker dependency or fundamental frequency dependency. In section 2.4 an overview of existing performance evaluation methods are given including subjective tests. Since the mentioned separation methods are often evaluated with different performance measures, a direct comparison is difficult or even impossible. For this reason, the target-to-masker ratio (TMR), which is a simple SNR measure in the frequency domain, is used for evaluation throughout this work.

Chapter 3

Model-Based Single Channel Source Separation

3.1 Introduction

Single channel source separation belongs to the class of underdetermined optimization problems. Mathematically, one variable is observed and the determination of N unknown hidden variables is impossible without further imposed constraints [3, Ch.4]. Throughout this work we assume $N = 2$. Figure 3.1 depicts the factorial model for the single channel source separation (SCSS) problem. The speech segments S_1 and S_2 are combined at each time instance τ to produce the speech mixture Y. Source separation is concerned with the reverse process. For the given speech mixture, the extraction of the source signals is of interest. Within this task, prior knowledge about the source signals is utilized for separation. As a constraint, the form of the individual source characteristics is pre-specified, regardless of the interference. These individual source characteristics can then be used as *prior models* for the sources. In this thesis, only speech signals are treated. Thus, a source refers to a human speaker. Assuming that there exists a defined rule, how to combine the set of source signals to yield the observation, the source separation becomes the problem of finding these signals.

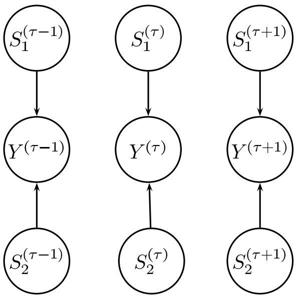


Figure 3.1: Source separation problem: The mixture Y is obtained by the combination of the two source signals S_i in every time step τ.

The inference in factorial models [93, 94] is a common problem in probabilistic models. Therefore, a hidden random variable z_i is defined for every source signal S_i. In this work a mixture of two sources is assumed, therefore, $i \in \{1, 2\}$. Then the goal of inference is the maximization of the joint likelihood $p(Y, z_1, z_2)$:

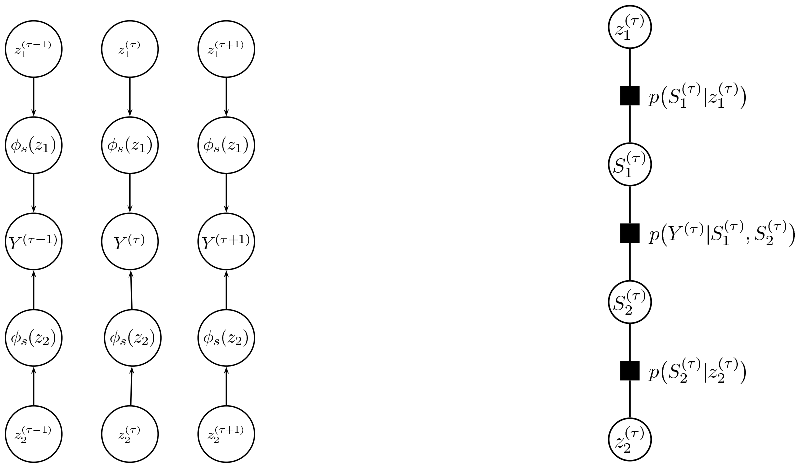
$$p(Y, z_1, z_2) = p(Y|z_1, z_2)\ p(z_1, z_2), \tag{3.1}$$

where $p(Y|z_1, z_2)$ is the conditional likelihood of Y given z_i and $p(z_1, z_2)$ is the joint prior probability. Separation aims at finding the most likely states z_1 and z_2 representing Y, which from its nature is a classification problem. As both, observation Y and source signals S_i are continuous, the learning process of all possible combinations of the associated continuous hidden random variables is hard. For that reason, observation or interaction models have been introduced [95, 96, 97, 17]. The interaction model provides rules how to combine the hidden variables in order to explain the speech mixture. Hence, observation models enable unsupervised methods to independently represent the sources during training. Therefore, statistical methods are employed. During this modeling process the continuous sources are discretized in $|Z_i|$ states, where each state z_i represents a specific class of source components. Here, $z_i \in Z_i$, where Z_i has cardinality $|Z_i|$. Thus, statistical models are trained in a generative manner to represent source characteristics. For separation, tracking, or decoding, the observation models are used to combine the independently assumed source models in a defined way into a factorial model [93] and thus explain the observed mixture. For each source model, a discrete hidden variable is defined, which evolves independently over time. Afterwards, the emission densities related to the state sequences of each hidden variable are employed for signal separation. Generally, the mean values μ_i^z of the emission densities for states z_i are utilized for this purpose. This entire process is illustrated in Figure 3.2(a), where emission densities $\phi_s(z_i)$ are explicitly depicted for better understanding. A factor graph representation of the intuitive/factorial source separation model of Figure 3.2(a) is shown in Figure 3.2(b). In this representation, the conditional dependencies of the signals and the hidden states are explicitly described. A comparison to Figure 3.1 shows that each continuous speech segment of the source signals is replaced by the emission densities of the speaker dependent model. For a particular time segment, the hidden variables draw specific emission densities in order to explain the observed speech mixture. To reconstruct the underlying source signals, the means of the emission densities are either directly used for signal syntheses or are utilized to derive masking signals. The speech mixture is weighted by these masking signals to finally synthesize estimates of the source signals.

In summary, model-based source separation can be divided into four building blocks:

- Source model training
- Separation or decoding
- Observation or interaction model
- Signal reconstruction
  - Direct synthesis
  - Masking signal estimation

In the sequel, these four modules will be discussed in more detail. At the beginning, a probabilistic formulation is developed for the source separation task. Afterwards, the observation models are discussed. Subsequently, we review three different masks for signal reconstruction. This chapter comprises detailed analysis of the source separation problem, the experiments addressing model complexity and model training, and the soft binary mask for signal reconstruction.



(a) Statistical source separation problem (b) Equivalent factor graph representation

Figure 3.2: Source separation problem using generative models. (a) The mixture Y is obtained by the combination of the means of the two emission densities $\phi_s(z_i)$ specified by their hidden variables z_i in every time step τ. (b) Factor graph representation: The conditional dependency of the source signal on the random variable is explicitly shown by the factor nodes (filled nodes). Circles denote variable nodes.

3.2 Inference in Factorial Models

Now, given the speaker dependent (SD) models and assuming that we have access to the state sequence chosen by the hidden variables z_1 and z_2 associated with each speaker, the joint distribution of the observation and the underlying source signals S_i for a particular instant of time is given as:

$$p(Y, S_1, S_2|z_1, z_2) = p(Y|S_1, S_2)\ p(S_1|z_1)\ p(S_2|z_2), \tag{3.2}$$

where z_i selects the proper speaker dependent speech segment S_i best representing the current observed variable. We assume all random variables $z_i \in Z$, where Z has the cardinality $|Z|$. The posterior distribution for Y given the unobserved hidden variables z_i can be found by marginalization over the underlying signal components as:

$$p(Y|z_1, z_2) = \int_{S_1}\int_{S_2} p(Y|S_1, S_2)\ p(S_1|z_1)\ p(S_2|z_2)\ dS_1 dS_2. \tag{3.3}$$

A factor graph [1] representation of the factorial HMM is shown in Figure 3.3. In the model, each hidden random variable is described by a variable node which is related to a Markov chain. As this model combines two HMMs, the model is called factorial HMM (F-HMM). Factor nodes are depicted as shaded nodes describing a local function. Edges are connecting factor nodes and variable nodes, if and only if the variable node is a function of the factor node. The connecting edges indicate direct conditional dependency. The prior distribution of each random variable is denoted by $p(z_i)$. Moreover, the transition probabilities $p(z_i^{(\tau)}|z_i^{(\tau-1)})$ model the relationship of a random variable between two consecutive time instances. Note, the explicit dependency between z_i and S_i, modeled as $p(S_i|z_i)$, is absorbed into factor nodes in the F-HMM.

The next section focuses on the computation of the observation likelihood. Therefore, only memoryless statistical models are considered. Hence, we assume non-informative transition

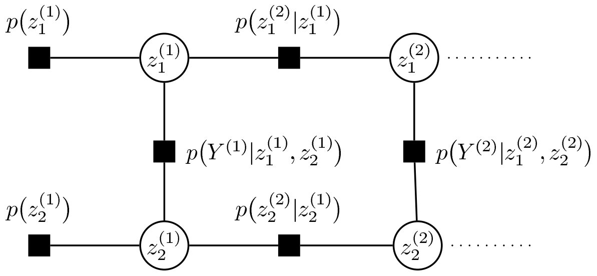


Figure 3.3: Factor graph representation of the factorial model. The mixture **y** is represented by two hidden variables z_i in every time step τ.

probabilities $p(z_i^{(\tau)}|z_i^{(\tau-1)})$, i.e. the probability mass is uniformly distributed in the transition matrix, in Figure 3.3. However, the observation likelihood computation also applies for first order Markov chains.

3.2.1 Observation Likelihood Computation

The aim of source separation is to compute the observation likelihood $p(Y|z_1, z_2)$ conditioned on the state sequences. To form an estimate of the component signals, either the minimum mean square error (MMSE) estimator $E(S_i|Y)$ defined as

$$E(z_i|Y, z_j) = \int z_i \; p(z_i|Y, z_j)\mathrm{d}z_i, \quad i \neq j; \quad i, j \in \{1, 2\} \tag{3.4}$$

or the maximum *a posteriori* (MAP) estimate has to be computed. The MAP can be found by the Bayes theorem using Eq. (3.3) as:

$$p(z_1, z_2|Y) = \frac{p(Y|z_1, z_2)\; p(z_1)\; p(z_2)}{p(Y)},$$

where $p(z_1)$ and $p(z_2)$ are assumed to be independent prior distributions. Thus, the most likely states can be found by:

$$\{z_1^\star, z_2^\star\} = \underset{z_1, z_2}{\operatorname{argmax}} \big[p(z_1, z_2|Y)\big]. \tag{3.5}$$

Assuming uniform priors $p(z_1)$ and $p(z_2)$ and neglecting the normalizing factor $p(Y)$ and $p(z_1, z_2|Y) \propto p(Y|z_1, z_2)$, we can further write $\{z_1^\star, z_2^\star\} = \operatorname{argmax}_{z_1, z_2} \big[p(Y|z_1, z_2)\big]$. Note, this simplification results in the maximum-likelihood solution.

Additionally, we assume that the density function conditioned on the states is deterministic and has the following property:

$$p(S_i|\; z_j) = \begin{cases} 1, & \text{if} \quad i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

This condition emphasizes the sparseness constraint of the MAP solution. At time step τ just one state can be active for each model. Additionally, introducing this assumption in Eq. (3.3) results in the following relation: $p(Y|z_1, z_2) = p(Y|S_1, S_2)$, where we represent S_i by the Gaussian mean or cluster center $\phi_s(z_i) = \mu_s(z_i)$ which has been drawn by the latent variable z_i. The computational complexity for the observation likelihood computation is

$\mathcal{O}(T\ |Z|^2)$. If the model of Eq.(3.1) is extended to a F-HMM, the joint probability changes to:

$$p(y,\{z\}) = \prod_{i=1}^{2} p(z_i^{(1)})\ p(y^{(1)}|\{z^{(1)}\}) \prod_{\tau=2}^{T} p(y^{(\tau)}|\{z^{(\tau)}\})\ p(z_i^{(\tau)}|z_i^{(\tau-1)}), \tag{3.7}$$

where $\{z\}$ denotes the wholes set of variables. In contrast to the model without dynamics, the F-HMM additionally includes time dependency by the transition probabilities.

For separation, the most likely state sequence is determined using the 2-D Viterbi algorithm [96]. The Viterbi search is an efficient way to compute maximum the joint state log-likelihood $w(\{z^{(\tau)}\})$ over the state $z_i^{(\tau)}$ and observation sequence $Y^{(\tau)}$ for $\tau = [1,\ldots,T]$. The recursion is given as:

$$\begin{aligned} &w(\{z^{(\tau)}\}) = \\ &\log\ p(Y^{(\tau)}|\{z^{(\tau)}\}) \max_{\{z^{(\tau-1)}\}} \big(\ \log\ p(z_1^{(\tau)}|z_1^{(\tau-1)}) + \log\ p(z_2^{(\tau)}|z_2^{(\tau-1)}) + w(\{z^{(\tau-1)}\})\ \big). \end{aligned} \tag{3.8}$$

For the F-HMM the Viterbi algorithm recursively tracks the most likely state sequence in a three dimensional state space, i.e. z_1, z_2, and τ. The best paths to all states at time τ are computed using the best paths to all states at time $\tau-1$. At the final step of the sequence, the state with greatest likelihood is utilized for back tracking, which extracts the most likely state sequence. An analysis of computational complexity reduction for this method is thoroughly discussed in chapter 4.

3.3 Observation Models for Factorial Models

In this section, commonly used observation models are discussed. Therefore, we assume co-channel speech, which is the linear instantaneous mixture of two speakers:

$$y(t) = s_1(t) + s_2(t) + \iota(t), \quad t = [1,\ldots,T], \tag{3.9}$$

where $s_i(t)$, with $i \in \{1,2\}$ is the respective speaker, $\iota(t)$ is an optional noise component, e.g. sensor and/or background noise, and T denotes the length of the utterance. Moreover, we consider the component sources to be combined at equal energy level. In the sequel we are neglecting $\iota(t)$. In the complex frequency domain the relationship between the component signals and the observed speech mixture is given as

$$Y = S_1 + S_2. \tag{3.10}$$

Note, we denote the complex spectrogram with upper case symbols, i.e. Y and S, the magnitude spectrogram with lower case bold symbols, i.e. $\mathbf{y}$ and $\mathbf{s}$, and the log-magnitude with upper case bold symbols, i.e. $\mathbf{Y}$ and $\mathbf{S}$. An equivalent representation in terms of magnitude and phase is given as:

$$\mathbf{y}^2 = \mathbf{s}_1^2 + \mathbf{s}_2^2 + 2\cdot\mathbf{s}_1\ \mathbf{s}_2\ \cos(\psi), \tag{3.11}$$

where ψ is the phase difference $\psi = \psi_2 - \psi_1$ between the source signals. A simplification models the relationship between the sources and the speech mixture by representing the phase term as an error $\nu = 2\cdot\mathbf{s}_1\ \mathbf{s}_2\ \cos(\psi)$:

$$\mathbf{y}^2 = \mathbf{s}_1^2 + \mathbf{s}_2^2 + \nu. \tag{3.12}$$

In the literature, various observation or interaction models have been defined within this framework for the posterior likelihood $p(y|\{z\})$ computation. In the next sections the additive, mixture-maximization, and the Algonquin observation model will reviewed.

3.3.1 Additive Model

In the additive model, the mixture is approximated in the squared-magnitude frequency domain by $\tilde{\mathbf{y}}^2 \approx \mathbf{s}_1^2 + \mathbf{s}_2^2$. For this model, the approximation error is given as $\nu = \mathbf{y} - \tilde{\mathbf{y}} = 2\mathbf{s}_1\ \mathbf{s}_2\ \cos(\psi)$. In general, it is assumed that source signals follow a Gaussian distribution. The density of the sum of two independent random variables results in the convolution of their probability densities. The convolution of two Gaussians is also Gaussian, hence, the posterior likelihood model is given as:

$$p(\mathbf{y}^2|\mathbf{s}_1^2, \mathbf{s}_2^2) = \mathcal{N}(\mathbf{y}^2; (\mathbf{s}_1^2 + \mathbf{s}_2^2), \sigma^2), \tag{3.13}$$

where $\mathcal{N}$ is the normal distribution and σ^2 is the variance of the approximation error ν. It can be shown that this approximation under the Gaussian observation model is indeed a minimum mean square error (MMSE) estimator. The optimality of the MMSE is given for an uniformly distributed phase [98, 28]:

$$E\{\nu\} = \int_{-\pi}^{\pi} (\tilde{\mathbf{y}} - \mathbf{y})\ \mathrm{d}\psi = \int_{-\pi}^{\pi} (2\mathbf{s}_1\ \mathbf{s}_2\ \cos(\psi))\ \mathrm{d}\psi = 0. \tag{3.14}$$

Moreover, the variance of the error can be computed by taking the expectation of the squared error $\sigma^2 = E\{\nu^2\}$. Solving this integral results in a variance of: $\sigma^2 = 2\ \mathbf{s}_1^2\ \mathbf{s}_2^2$.

3.3.2 Mixture-Maximization Model

In the logarithmic frequency domain, the speech mixture can be approximated by the mixture-maximization (*mixmax*) [95] operator as:

$$\mathbf{Y} \approx \max(\mathbf{S}_1, \mathbf{S}_2), \tag{3.15}$$

where the maximum operation is carried out elementwise over each dimension. The motivation for this operator stems from the visual inspection of speech signals in individual frequency bands [19], where frequency cells rarely contain significant energy across long time intervals. Thus, the *mixmax* approach results from the fact that speech is sparsly distributed in time-frequency representations [3]. Recently, Radfar et al. [99] showed that Eq. (3.15) is a nonlinear minimum mean square error estimator under the assumption of uniformly distributed phases of the source signals. For the observation likelihood computation we assume a Gaussian distributed model as:

$$p(\mathbf{Y}|\mathbf{S}_1, \mathbf{S}_2) = \mathcal{N}(\mathbf{Y}; \max(\mathbf{S}_1, \mathbf{S}_2), \sigma^2). \tag{3.16}$$

σ^2 is the variance of the approximation error or of an optional observation noise ι.

In a probabilistic model, the speech segments are replaced by the means of the emission densities $\phi_s(z_i)$. Thus, they are no longer constant vectors but probability density functions $p_s(z_i)$. Nadas et al. [95] have shown that the approximation of Eq. (3.16) in the probabilistic representation is given as the multiplication of the two independent random variables. This results in a multiplicative relation of their cumulative distribution function Φ:

$$\Phi_y(z) = \Phi_s(z_1)\Phi_s(z_2), \tag{3.17}$$

where Φ is the cumulative distribution function. By differentiation of $\Phi_y(z)$ the probability density distribution of the speech mixture can be found as:

$$p_y(\{z\}) = \phi_s(z_1)\Phi_s(z_2) + \Phi_s(z_1)\phi_s(z_2), \tag{3.18}$$

where $p(Y|z_1, z_2) = p_y(z)$. Using Eq.(3.18), an efficient way the compute the posterior likelihood has been found. The *mixmax* model has been further extended from single Gaussians to multiple Gaussians per state, i.e. GMMs, by [87, 100].

3.3.3 Algonquin Model

The Algonquin model was initially introduced for robust speech recognition [97] in noisy environments. This model defines how speech and noise interact. For speech separation, this model was extended by Kristjansson et al. [101]. The Algonquin model describes the relation between the source signals and the speech mixture by taking the logarithm of Eq.(3.12):

$$\mathbf{Y} = \mathbf{S}_1 + \ln(1 + \exp(\mathbf{S}_2 - \mathbf{S}_1)) + \varepsilon, \tag{3.19}$$

where ε is assumed to be normal distributed random noise. The posterior likelihood is finally assumed to be Gaussian:

$$p(\mathbf{Y}|\mathbf{S}_1, \mathbf{S}_2) = \mathcal{N}(\mathbf{Y}; \mathbf{S}_1 + \ln(1 + \exp(\mathbf{S}_2 - \mathbf{S}_1)), \Sigma^2), \tag{3.20}$$

where Σ^2 is the variance of the normal distribution. Since, the true posterior is non-Gaussian due to the non-linear relation in Eq.(3.19), a linearization using a first order Taylor series expansion at a point q_0 is carried out. Therefore, $g(q) = \mathbf{S}_1 + \ln(1 + \exp(\mathbf{S}_2 - \mathbf{S}_1))$ and $q = [\mathbf{S}_1^T, \mathbf{S}_2^T]^T$ are introduced, as well as $g'(q)$, as the first derivative of $g(q)$ evaluated at q. Then, the linearization of the posterior likelihood follows as:

$$p_l(\mathbf{Y}|\mathbf{S}_1, \mathbf{S}_2) = \mathcal{N}(\mathbf{Y}; g(q_0) + g'(q_0)(q - q_0), \psi) \tag{3.21}$$

For the maximization of the function in Eq. (3.21) the iterative Newton algorithm is utilized.

3.4 Signal Reconstruction

Once the state sequence for each speaker is estimated, the state emission probabilities for each state are used to synthesize the underlying speech signals. Usually, these prototype signals of each speaker are used to derive a masking signal [102]. The extracted masks are finally used to weight the speech mixture in order to recover the component signals as:

$$\hat{S}_i^{(\tau,d)} = Y^{(\tau,d)} \cdot m_i^{(\tau,d)}, \quad i \in \{1, 2\}. \tag{3.22}$$

Where $m_i^{(\tau,d)}$ is the mask for source i. The mask is basically a weighting of each time-frequency cell (τ, d) of the speech mixture. By this operation, the source signal is recovered. Alternatively, the estimated means of the emission densities can be directly synthesized using the phase of the speech mixture:

$$\hat{S}_i^{(\tau)} = \phi(z_i^{(\tau)}) \cdot \angle Y^{(\tau)} \tag{3.23}$$

However, these reconstructed signals are not well suited to make them direct audible. Due to the missing fine structure over frequncey and smoothness across time, the intelligibility is very low. In the literature different masks have been proposed for signal reconstruction. Reddy et al. [103, 104] proposed a minimum mean square estimator for separation. They used the final estimate as masking signal for reconstruction. Alternatively, a mask similar to the Wiener filter solution [92] can be used for separation and reconstruction. We denote this method as soft mask in the sequel.

In this section we investigate three different masks: (i) the binary mask, (ii) the soft mask, and (iii) the soft binary mask.

Binary Mask: Recently, Wang [18, 19] suggested the ideal binary mask as computational goal of auditory scene analysis. Indeed, Roweis [69] had shown that two speakers are rarely simultaneously active in a time-frequency cell of a high resolution time-frequency representation. This directly results in the independence assumption of the component signals, due to the sparse distribution of speech in such a representation. The target ideal time-frequency binary mask assigns a "1" to a time-frequency cell if the target energy is greater than the interference energy and "0" otherwise. For the interference, the ideal binary mask can be computed in a similar way. Thus, the binary masks are complementary, i.e. $\mathrm{BM}_1 = \overline{\mathrm{BM}_2}$. To compute the ideal binary mask, the component signals must be available in advance, which is in general an unreasonable assumption. The goal of any SCSS algorithm is the estimation of the binary mask, which is in the best case identical with the ideal binary mask. Therefore, the mixture maximization (*mixmax*) approach [95], i.e. the element-wise maximum operator applied on a time-frequency representation, is employed. This justifies the use of the *mixmax* approach as interaction model for binary mask signal reconstruction.

Soft Mask: The binary mask makes a hard decision in exclusively assigning time-frequency cells. Accordingly, if a frequency cell is allocated to the false speaker, the decision is totally wrong. To overcome this issue, a soft mask can be defined. The soft mask assigns a continuous value between zero and one to each time-frequency cell and is defined as follows:

$$m_1^{(\tau,d)} = \frac{\mathbf{s}_1^{(\tau,d)}}{\mathbf{s}_1^{(\tau,d)} + \mathbf{s}_2^{(\tau,d)}}, \quad \forall\ \tau, d. \tag{3.24}$$

Here, $\mathbf{s}_1$ and $\mathbf{s}_2$ are the magnitude spectra of the respective target and interference signals. For source separation, these signals are replaced by their estimates, i.e. the emission density means. The soft mask of the target is m_1 and $m_1 + m_2 = 1$.

Soft Binary Mask: The soft binary mask is somehow a compromise between the binary and soft masks. The introduction of this mask was motivated by the observation likelihood approximation of Reddy and Rennie [103, 105]. They assumed that only one speaker contributes significant energy to a time frequency cell. Thus, the observation likelihood can be approximated by $p(\mathbf{y}|\{z\}) \approx p_{z_i}(\mathbf{y}|z_i)$. The posterior expected value can be finally represented as either $E(\mathbf{s}_i = \mathbf{y}|z_i) \approx \mathbf{y}$ or $E(\mathbf{s}_i < \mathbf{y}|z_i) \approx \min(\mathbf{y}, \mathbf{s}_i)$. This idea can be equivalently used for mask estimation. The soft binary mask assigns a "1" to the time-frequency cell if the target energy is greater than the interference. Otherwise, the minimum of either mixture or target energy is assigned to the cell:

$$m_i^{(\tau,d)} = \begin{cases} 1, & if\ \mathbf{s}_i^{(\tau,d)} > \mathbf{s}_j^{(\tau,d)} \\ \frac{min(\mathbf{y}^{(\tau,d)}, \mathbf{s}_i^{(\tau,d)})}{\mathbf{s}_i^{(\tau,d)} + \mathbf{s}_j^{(\tau,d)}}, & \text{otherwise,} \end{cases} \tag{3.25}$$

where $i, j \in \{1, 2\}$ and $i \neq j$. With the soft binary mask we try to recover the masked residual energy.

3.5 Experimental Results

In all source separation experiments the *mixmax*-approach is used as interaction model. Moreover, temporal dependencies are neglected. Hence, for separation the factorial-max VQ

(FM-VQ) which operates in the log-frequency domain is employed. A VQ model is trained for every speaker independently, using the k-means algorithm. Since all interaction models show on average a similar performance, differences among them are out of the scope of this thesis. From the description of the source separation modules, the following questions will be discussed in the experiments:

1. What is the optimal number of used codewords/states to model speaker characteristics? Is it worth to train models until convergence in case of the k-means?

2. Is there a dependency between the used number of iterations for model training and the separation performance?

3. Is there an advantage using statistical models instead of a large dictionary built on using template speech segments?

4. What is the best reconstruction method for a given codeword/state sequence?

Model complexity and performance in VQ models: In the first experiments, the impact of the number of codewords used to model speaker specific characteristics are studied. Therefore, the speaker space was clustered in $Q = [20, 50, 100, 200, 300, 500, 1000, 1500, 2000]$ non-intersecting cells. Separation results depending on the used number of codewords are shown in Figure 3.4. The labels indicate the number of states or codewords $Q \equiv |Z|$. Certainly, separation performance increases almost monotonically with the number of components and starts to saturate above $Q = 1000$. Intuitively we expect that an increasing number of codewords also increases separation performance until performance converges to the result of the ideal binary mask. It seems plausible, the more components we use the better we can explain specific speaker characteristics. But it is also obvious that the more space is occupied by a speaker the higher the risk of an overlap with a competing speaker, which results in ambiguities.

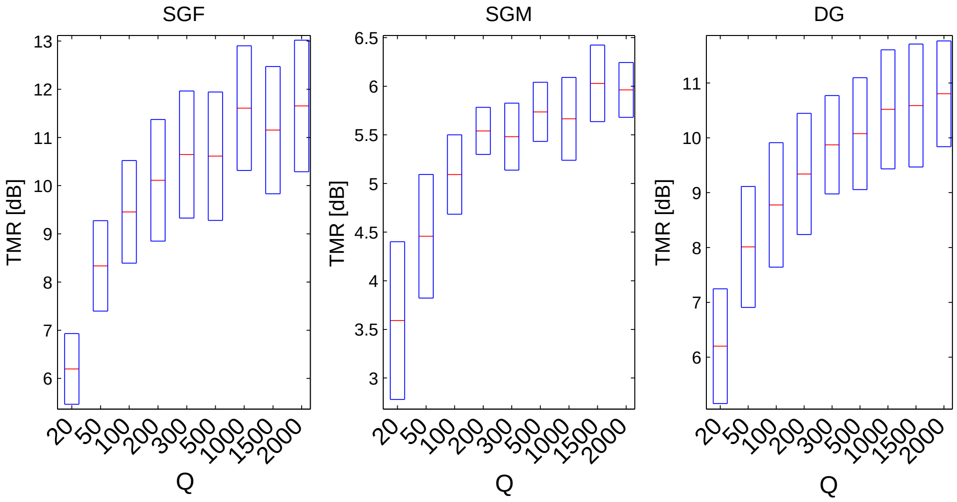


Figure 3.4: Factorial-max VQ single channel separation performance. The dependency between separation performance and the used number of codewords (Q) to model speaker characteristics are investigated.

Investigation of used iterations for VQ model training: Furthermore, we found a relationship between the number of used iterations to train a VQ model and the separation performance. In the experiment, the number of components is fixed to $Q = 500$ and the training of the k-means algorithm always had been initialized with the same parameter setting. Training was stopped after $iT = [1, 5, 10, 20, 30, 50, 100, 150, 200]$ iterations, respectively. The separation performance for the 3 different cases is depicted in Figure 3.5. The label at the bottom of each subplot indicates the number of iterations used for training.

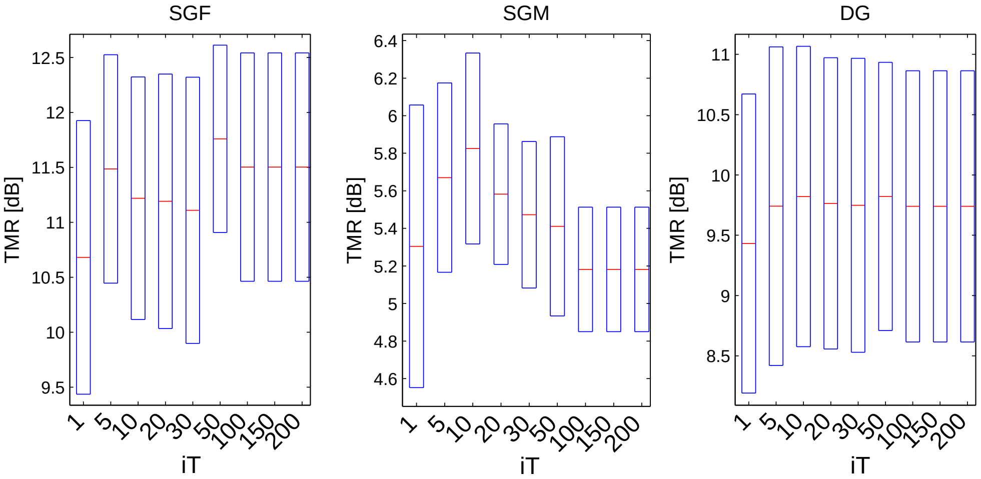


Figure 3.5: Factorial-max VQ separation results. In this experiment the number of components is fixed, varying just the number of iterations (iT) for model training.

Indeed, we observed that there is an impact on the separation performance, whereas the differences are not significant for the SGF and the DG case. However for the SGM case with 10 iterations, the achieved TMR is approximately 5.8 dB, whereas for 200 iteration the TMR reduces slightly below 5.2 dB. For the other two cases, the variations between minimum and maximum achieved TMR is also around 0.5 dB. However, in these cases, there is no linear relation between the used number of iteration and TMR performance. Moreover, we see that the training saturates and stops at a stationary point.

Template versus statistical models: The next experiment tries to answer the third question, whether a statistical model can be replaced by a dictionary containing just speaker specific templates. If this hypothesis is true, the question remains, if we can find the best dictionary template observing just the speech mixture. The results shown in Figure 3.6 answer the first question. Results for template dictionaries using $|D| = [500, 6000, 12000]$ entries are illustrated. Speech templates are log-spectral segments which have been randomly drawn from the training data. A comparison to Figure 3.4 shows no significant improvement using templates. Moreover, the computational complexity for inference of the template based method using 6000 or 12000 templates is significantly higher, i.e. $\mathcal{O}(|D|^2\ T)$. For the FM-VQ using template models, the iterated conditional modes algorithm, introduced in chapter 4, section 4.5 to approximate the observation likelihood was employed. This reduces the computational complexity significantly.

Signal reconstruction experiments: To evaluate the performance of the introduced signal reconstruction methods on the specified test database we use the reference component

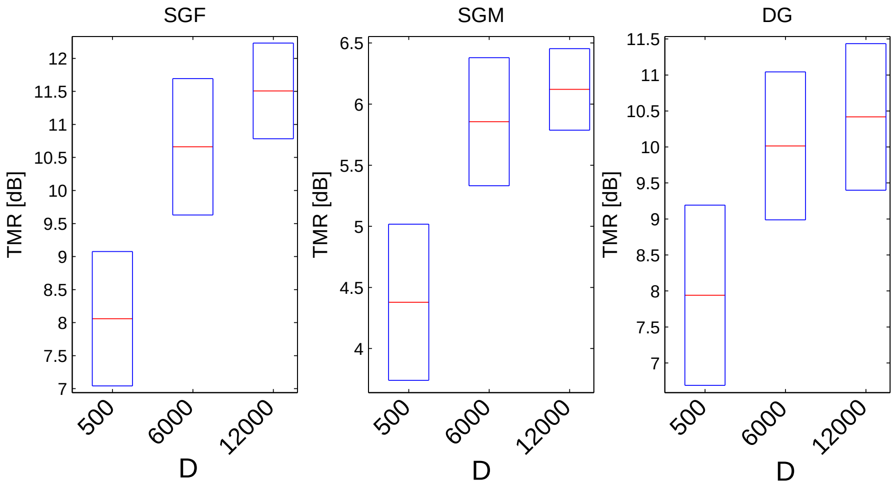


Figure 3.6: Factorial-max VQ separation results employing a speaker specific template dictionary. The label indicates the cardinality of the template dictionary D employed for separation.

signals to extract the respective ideal masks. This leads to the (i) ideal binary mask (iBM) for the binary masks, (ii) the ideal soft mask (iSM) for the soft masks, and (iii) the ideal soft binary mask (iSBM) in case of the last introduced mask. Figure 1.3 shows the spectrogram, orthographic transcription and time-domain signal of the sentence “lay white in x zero soon” spoken by speaker FE1. The speaker FE1 is mixed at equal level with speaker FE2 uttering the sentence “lay white with y two again”. The extracted ideal masks for speaker FE1 are shown in Figure 3.7. A “1” is represented by a white pixel whereas a “0” is represented by a black pixel. Values between “1” and “0” are depicted with the appropriate gray color. Comparing the masks to each other, they all recover mainly the same time-frequency regions. In addition, there is a strong correlation of the white mask regions and high energy regions of the clean FE1 utterance, shown in Figure 1.3.

The results for the three different ideal masks in terms of TMR for the whole test database are shown in Figure 3.8. The TMR is plotted separately for the SGF, SGM and DG case.

To verify, if the estimated masks show the same overall behavior as their ideal masks, the estimated state sequence of the FM-VQ model is used for mask estimation. The results in TMR for the reconstructed signals are shown in Figure 3.9. Since we can not access the underlying signals for mask estimation but just their statistical representation, we can expect that the differences between the different reconstruction methods should decrease. Indeed, for the SGF case we observe this behavior, but for the SGM and the DG cases the difference in the means between the masks remains (see figure 3.9). Only for the SGF case, the SBM achieves slightly higher performance compared to the other masks. A comparison of the ideal mask results (Figure 3.8) and their approximations (Figure 3.9) shows that the TMR for all cases drops. We conclude that the main decrease of the TMR between ideal and estimated masks is due to the missing details in the statistical representation of the true source signals at every instance of time.

In appendix B a performance analysis of the three discussed masks is provided for the application of automatic speech recognition.

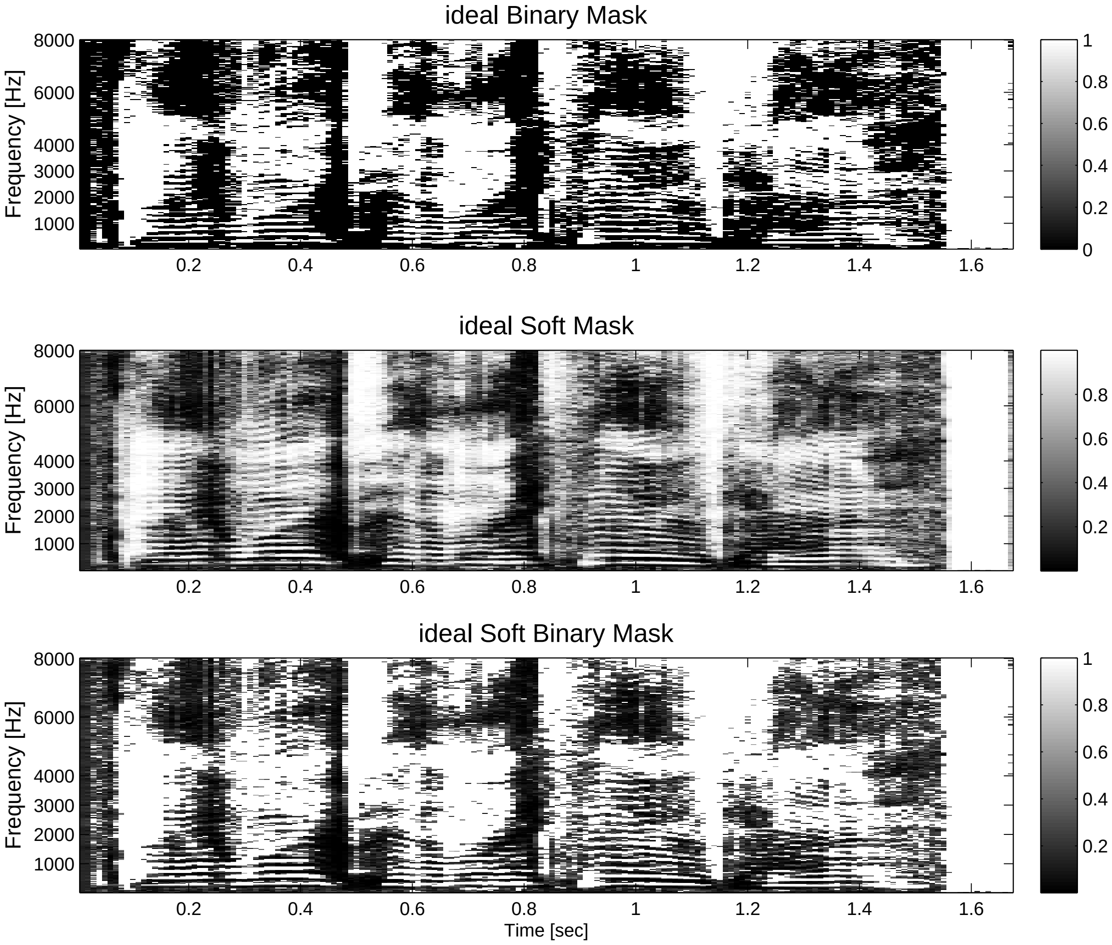


Figure 3.7: Ideal masks of target speaker FE1. **Top:** ideal binary mask. **Middle:** ideal soft mask. **Bottom:** ideal soft binary mask. White pixels indicate a “1” in the mask and black pixels a “0”.

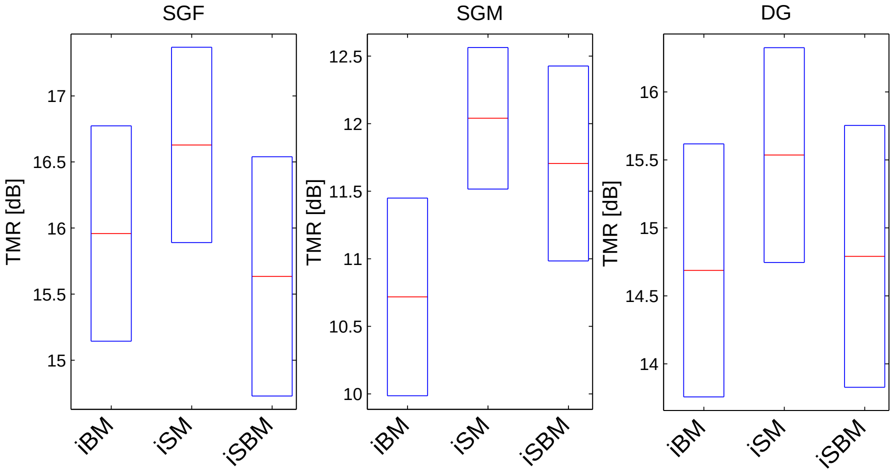


Figure 3.8: Single channel separation results of two speakers using the ideal binary mask (iBM), ideal soft mask (iSM), and the ideal soft binary mask (iSBM).The TMR results in [dB] are illustrated for the SGF, SGM, and DG case.

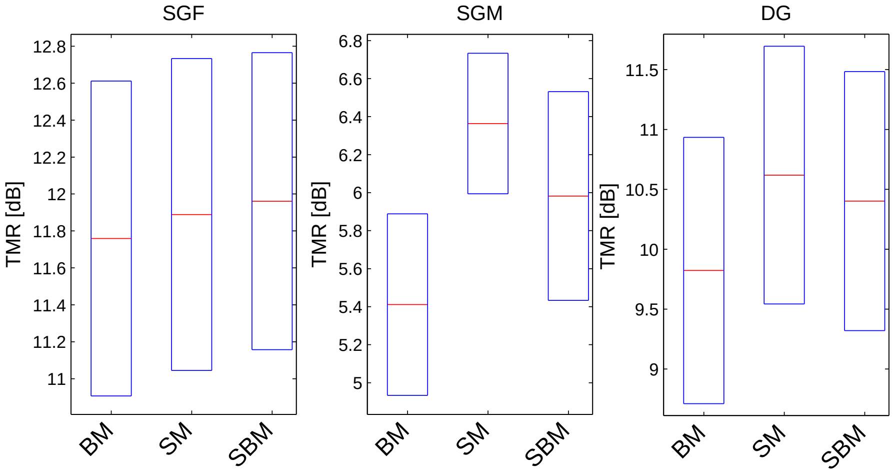


Figure 3.9: FM-VQ single channel separation results employing the binary mask (BM), soft mask (SM), and the soft binary mask (SBM) for signal reconstruction.The TMR results in [dB] are illustrated for the SGF, SGM, and DG case.

3.6 Conclusion

This chapter introduced speaker dependent statistical models for SCSS. Moreover, the whole statistical framework for factorial memoryless models and factorial HMMs was discussed. Specifically, all building blocks for source separation have been introduced. All three state-of-the-art interaction models, namely, the linear, the mixmax, and the Algonquin models have been described. In the experiments, we addressed the issues of model complexity and model training. For increasing number of states, also the performance increases until it saturates. Furthermore, we observed that there is no need to train a VQ model until convergence. Quite the contrary is true as for the same gender male case, the TMR is decreasing with increasing number of iterations used for training. Finally, different signal reconstruction methods to synthesize the source signals are discussed. Particularly, the soft binary mask was proposed as compromise between the exclusive energy assignment of the binary mask and the continuous valued soft mask. The ideal masks have been compared to each other as well as to the estimated ones. Hence, we believe that the choice of mask for signal reconstruction is application dependent.

Chapter 4

Computational Complexity in Factorial Models

4.1 Introduction

The computational complexity due to the factorial nature of model-based source separation methods significantly restricts their suitability for close to real-time applications. In particular, for separation purposes, the spanned space of two sources is $\mathcal{O}\left(Q^2\right)$, where Q denotes the cardinality $|Z|$ of a speaker dependent statistical model. This corresponds to the complexity to compute the observation likelihood for one time step. A typical factorial-VQ structure is shown in Figure 3.3 and has been discussed in chapter 3.

However, once the state sequence associated to each speaker is at hand, a binary, continuous, or soft binary mask can be found for each speaker. Finally, the mixture is filtered by the masks and the underlying signals are estimated and reconstructed (see chapter 3, section 3.4).

This chapter deals with the efficient state likelihood estimation at every time step. Hence, the goals is to approximate the MAP solution at every single time step without computing the full observation likelihood. Therefore, we start with a discussion on the full observation likelihood computation for factorial models. Afterwards, in order to reduce the computational burden, heuristics to approximate the observation likelihood will be introduced and discussed. To this end, we either adapt beam search or propose the use of the iterated conditional modes (ICM) algorithm, which is a special case of the Gibbs sampling method [106]. The methods are compared to the hierarchically structured VQ model proposed in [36]. No comparison will be made to the brunch and bound method [17], as this method did not achieve a significant reduction of complexity in our experiments. All three algorithms reduce the computational costs by two orders of magnitude compared to full search, whereas the implications on the separation performance is minimal. Note, throughout this chapter we use the *mixmax* observation model [95] of chapter 3, section 3.3.2 as interaction model.

Additionally, we integrate these approximations in factorial-max HMMs (FM-HMMs). All methods are evaluated in terms of computational complexity and performance, which is evaluated by both, the achieved score in the likelihood-state plane and the obtained target-to-masker-ratio (TMR) during separation. Some conclusions will finalize this chapter. A short version of this chapter with different statistical models has recently been published in [107].

4.2 Full Observation Likelihood Computation (FS)

In order to perform source separation we are interested to find the most probable state sequence of each hidden variable for a given observation sequence. For the moment let us assume a factorial model without dynamics and uniformly distributed prior probabilities. In this case, for a given observation sequence the sequence of individually most likely states is identical to the most likely state sequence. Note, this is not true in general. To form an estimate of the component signals, the most likely state sequences have to be determined by computing the maximum *a posteriori* (MAP) estimate (see chapter 3):

$$\{z_1^\star, z_2^\star\} = \underset{z_1,z_2}{\operatorname{argmax}} \left[p(z_1, z_2|\mathbf{Y})\right].$$

As in general, large models are trained (e.g. $Q = 500$ Gaussian components for a Gaussian mixture model (GMM) or codewords for an VQ) to capture acoustic properties of each speaker, the computation of the joint posterior is $\mathcal{O}(Q^2)$ which is a bottleneck of this approach for real world applications. This is caused by the factorial nature of the model, which necessitates the computation of all state combinations of the two speaker models. For example, the computation of $p(\mathbf{Y}|z_1, z_2)$, given two speaker models each of cardinality 500, requires $500 \times 500 = 250000$ operations to evaluate all state combinations. This quantity has to be computed for every single time step. Hence, the computational complexity is $\mathcal{O}(T\ Q^2)$ for T time steps and Q states of the speaker models. Therefore, approximation techniques for efficient observation likelihood estimation are necessary to make this method applicable for close to real-time applications. In the following, we propose two methods, namely the fast beam search and Gibbs sampling, and compare them to the fast likelihood estimation method. As a focal point, we discuss the iterated conditional modes algorithm which is a greedy approximation of Gibbs sampling.

4.3 Fast Likelihood Estimation (FLE)

In order to alleviate the intensive task of state likelihood computation, Bocchieri [108] suggested vector quantization (VQ) of the input feature vector to identify a subset of Gaussians belonging to a particular VQ entry. Based on this work, Rennie [36] proposed a band quantization of the Gaussians for modeling the acoustics. For this purpose, he uses diagonal covariance matrices in the GMMs. Thus, all feature dimensions are independent and therefore all Q Gaussians can be efficiently approximated by $k \ll Q$ Gaussians. Each of the Q Gaussians is associated with one of the k Gaussians using the mapping function $M(z_i)$ as follows: $\hat{p}(\mathbf{S}_i|z_i) = \prod_d \mathcal{N}(\mathbf{S}_i; \mu_{M(z_i)}, \sigma^2_{M(z_i)})$, where $\hat{p}(\mathbf{S}_i|z_i)$ acts as surrogate of $p(\mathbf{S}_i|z_i)$ and the product ranges over all frequencies d. Hence, K Gaussians are represented by the smaller number of k Gaussians, where $Q = k \cdot K$. The k Gaussians are selected such as to minimize the KL-distance:

$$D\Big(\sum_{z_i} p(z_i)\ p(\mathbf{S}_i|z_i)\ ||\ \sum_{z_i} p(z_i)\ \hat{p}(\mathbf{S}_i|z_i)\Big) \tag{4.1}$$

This approach uses a tree like structure for each speaker dependent (SD) model. The top level consists of k Gaussians. To each Gaussian of the top level, K Gaussians are associated. During inference, the top level states of the SD models are combined and the most likely state combination is estimated. Subsequently, the K states associated to the most likely states of the top level are combined to a factorial model and are finally used to find the most likely states. Thus, the most likely states are found in a sequential manner. Our implementation correlates better with the one, introduced by Bocchieri [108]. Thus, we use the k-means algorithm to cluster the Q codewords into k clusters. For decoding, the most likely state for each speaker out of the k coarse-grained categories is first determined, requiring $\mathcal{O}\left(k^2\right)$ evaluations. Once, we have those categories, we evaluate $z_i^\star$ out of on average K bases for each speaker in the next level. Using this technique, the computational complexity can be reduced to $\mathcal{O}\big(T(k^2 \ + K^2)\big)$ operations.

4.4 Beam Search Technique (BS)

An efficient way to speed up the MAP estimation of Eq. (3.7) is to apply beam search (BS), which is used during the Viterbi decoding [109, 106] in HMMs. The Viterbi decoder aims to find the most probable sequence of hidden states in an HMM for given observations, which is an efficient way to maximize the joint distribution $w(z^{(\tau)})$ over the state sequence $z^1, \ldots, z^T$ for a given observation sequence $\mathbf{S}^1, \ldots, \mathbf{S}^T$:

$$w(\{z^{(\tau)}\}) = \max_{z^{(1)},\ldots,z^{(\tau-1)}} \log\ p(\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(\tau)}, z^{(1)}, \ldots, z^{(\tau)}). \tag{4.2}$$

For first order Markov processes (HMMs), this expression can be factored and the computation can be carried out recursively for every instant of time as:

$$w(z^{(\tau)}) = \log\ p(\mathbf{S}^{(\tau)}|z^{(\tau)})\ + \max_{z^{(\tau-1)}} \left[\log\ p(z^{(\tau)}|z^{(\tau-1)}) + w(z^{(\tau-1)})\right] \tag{4.3}$$

Finally, the most probable hidden state sequence can be found using backtracking. The Viterbi algorithm reduces the computational complexity in HMMs compared to a naive implementation significantly from $\mathcal{O}(T \cdot Q^T)$ to $\mathcal{O}(T \cdot Q^2)$, as it scales linearly with the sequence length, in contrast to the exponential scaling of the naive implementation.

For many real world applications, the number of states of an HMM is too large for an efficient execution of the Viterbi algorithm. Therefore, a further reduction of complexity can be achieved by the introduction of a beam search [51], as we have already proposed in [110]. Thus, the maximum over the states at step $\tau - 1$ is determined $P^{(\tau-1)} = \max_q w(z^{(\tau-1)})$. This value defines a dynamical threshold $\theta = P^{(\tau-1)}/G$, where G is an appropriately chosen constant. Finally, all states below the threshold θ are eliminated, e.g. $w(z^{(\tau-1)}) = -\infty$ if $w(z^{(\tau-1)}) < \theta$ and only the reduced set of states is further used to determine the log-likelihood $w(z^{(\tau)})$, using also the transition information $p(z^{(\tau)}|z^{(\tau-1)})$. As alternative, the N most likely states can be accepted as *survivors*, instead of determining and comparing to a threshold. The advantage of this method is a constant reduction of complexity.

For source separation however, two independent HMM chains are employed to explain the mixture observation. The factor graph representation of the FM-HMM is illustrated in chapter 3, Figure 3.3. In contrast to the FM-VQ, the FM-HMM additionally incorporates time dependencies. This is illustrated by the connecting edges of the hidden variables in Fig. 3.3. In contrast to an HMM, the transition search space of an FM-HMM is increased to the power of four, e.g. Q^4. This is exemplified in Eq. (3.8). An illustration of the Viterbi search navigating through the 3-dimensional space is shown in [95]. For both beam search options, the state likelihood computation is not reduced. However, beam search has a great impact on the number of state transition computations.

Fast beam search: As GMMs and VQs do not belong to the family of first-order Markov processes, the beam search (BS) is not applicable and the full observation likelihood has to be computed.

We adapt the BS and apply it for memoryless statistical models (GMM and VQ). Therefore, we utilize the continuity property of speech, i.e. the energy in each frequency band of a magnitude spectrogram changes slowly over time. Additionally, we extract the spectrum $\mathbf{Y}^{(\tau-1)}$ with a time overlap of 50 percent – hence, at least half of the information contained in the previous mixture segment $\mathbf{Y}^{(\tau-1)}$ is also contained in the current $\mathbf{Y}^{(\tau)}$. The frequency representation of a short segment of speech mainly comprises contributions of the vocal folds (fine spectral structure) and the vocal tract (coarse spectral structure). Because of the mechanical properties, the vibration of the vocal folds and the vocal tract change continuously over time. Thus, speech can be represented as a slowly time varying process. Motivated by this property, we model speech as a martingale process [111]. For the observation likelihood computation, we apply this martingale theory in order to make a prediction of the most promising states from time step $\tau - 1$ to the next time step τ. Therefore, we model speech as a stochastic process $z^{(\tau-1)}$ and assume that the two martingale properties are met

$$\mathrm{E}(|z^{(\tau-1)}|) < \infty,$$

where the expectation of the absolute value of $z^{(\tau-1)}$ is bounded. The expected conditional probability of a random variable at time τ, given all previous observations is:

$$\mathrm{E}(z^{(\tau)}|z^{(1)}, \ldots, z^{(\tau-1)}) = z^{(\tau-1)}.$$

According to the martingale theory, the best guess we can make for a future time step, given the present and the past, is the present time step.

Bearing in mind the martingale theory, we can formulate the beam search for memoryless statistical models. Therefore, we specify N, the number of surviving states or VQ bases, i.e. the beam width. Furthermore, at step $\tau = 1$, using Eq. (3.5), we compute as initialization the full posterior likelihood and get $\{z_1^\star, z_2^\star\}$ the most likely state for each speaker model. Given the most probable states, the most similar states at the next time step are selected for z_1 and z_2, computing the posterior as:

$$p(z_1^{(\tau)}|\mathbf{Y}^{(\tau)}, z_2^{\star,(\tau-1)}) = \mathcal{N}(\mathbf{Y}^{(\tau)}; \max(\mu(z_1), \mu(z_2^{\star,(\tau-1)})), \Sigma), \tag{4.4}$$

where $\mu(z_i)$ are the state means of the random variable z representing speaker i. Here, we compute the likelihood of the first model being in state z_1 conditioned on the observation and the most likely state $z_2^\star$ of the second model and vice versa for the second model. Subsequently, we sort the likelihoods $p(z_1^{(\tau)}|\mathbf{Y}^{(\tau)}, z_2^{\star,(\tau-1)})$ and $p(z_2^{(\tau)}|\mathbf{Y}^{(\tau)}, z_1^{\star,(\tau-1)})$ in ascending order and specify a reduced set of $Q_i^{\prec}$ states, containing the N best matching states for each speaker used at τ as

$$\hat{p}^{(\tau)}(\mathbf{S}_i|z_i) = p(\mathbf{S}_i|z_i^{\prec}), \quad i \in \{1, 2\}, \tag{4.5}$$

where $Q_i^{\prec} \subseteq Q_i$ and $z_i^{\prec} \in Q_i^{\prec}$. This equation shows that Eq. (3.6) becomes dependent on time and that we only have ones where $z_i^{\prec} \in Q_i^{\prec}$. Hence, for time step τ, only the N most likely states, determined at time step $\tau - 1$, are considered. Using this fast beam search procedure, the computational complexity can be reduced from $\mathcal{O}(T\ Q^2)$ to $\mathcal{O}(Q^2 + (T-1)\ N^2)$. In the experiments, we will refer to this method as fast beam search (FBS) method.

This algorithm can be generalized in a way that, at time step $\tau - 1$, the M-best states are considered, instead of the most likely one. Therefore, Eq. (4.4) is generalized and the most likely states for time step τ are determined, conditioned on the M-best states of random variable z_i at time step $\tau - 1$. This step is carried out for each speaker independently. The generalized fast beam search method is summarized in algorithm 1.

```
 1: **Input:** $\mathbf{Y}$, VQ bases for speaker 1 and 2
 2: **Output:** $\{z^\star\}$
 3: $i \in \{1, 2\}$
 4: **for** $\tau \leftarrow 0$ **to** $T - 1$ **do**
 5:    **if** $\tau = 0$ **then**
 6:       Determine: $\{z_1^\star, z_2^\star\} = \text{argmax}_{z_1, z_2} \left[p(z_1, z_2|\mathbf{Y})\right]$
 7:    **else**
 8:       Sort descend $p(z_i^{(\tau)}|\mathbf{Y}^{(\tau)}, z_j^{\star,(\tau)}) \quad j \in \big(\{1, 2\} \setminus i\ \big)$
 9:       **for** $m \leftarrow 1$ **to** $M$ **do**
10:          Find: $p\big(z_1^{(\tau),m}|\mathbf{Y}^{(\tau)}, z_2^{(\tau-1),m}\big)$
11:          Find: $p\big(z_2^{(\tau),m}|\mathbf{Y}^{(\tau)}, z_1^{(\tau-1),m}\big)$
12:       **end for**
13:    **end if**
14:    Select $N$ best states of line 10 and 11: $z_i^{\prec} \in Q_i^{\prec}$ and $Q_i^{\prec} \subseteq Q_i(1 : N)$
15:    New Model: $\hat{p}^{(\tau)}(\mathbf{S}_i|z_i) = p(\mathbf{S}_i|z_i^{\prec})$
16:    Determine: $\{z_1^\star, z_2^\star\} = \text{argmax}_{z_1^{\prec}, z_2^{\prec}} \left[p(z_1^{\prec}, z_2^{\prec}|\mathbf{Y}^{(\tau)})\right]$
17: **end for**
```

Algorithm 1: Generalized Fast Beam Search Algorithm.

4.5 Gibbs Sampling (GS)

In this section, we investigate approximative inference methods, based on numerical sampling to speed up the likelihood computation. These so called Markov chain Monte Carlo (MCMC) [106, 93] methods use the posterior distribution to evaluate likelihoods, supposing that the direct likelihood computation is too complex. MCMC methods are in general used to make inference in probabilistic models. Thus, expectations, or in practice, likelihoods are computed in order to update model parameters in the maximization step. In particular, we employ the Gibbs sampling technique for approximative likelihood computation in factorial models for decoding.

The basic idea for the Gibbs sampling method is as follows [106]: We assume a certain distribution $p(\{z\}) = p(z_1, z_2, z_3)$ and some initial state of the variables. At each step, the Gibbs sampling procedure replaces one variable, while it keeps the others constant. The replacement is carried out by drawing a value for the selected variable from the distribution conditioned on the values of all other variables. This step is carried out for all variables, either by cycling through them in a specified order or randomly. For the above sample distribution with three variables and for step m of the algorithm, the procedure at the beginning computes $p(z_1|z_2^m, z_3^m)$. Thereafter, the variable z_1^m is replaced by z_1^{m+1} and we conditionally sample for $p(z_2|z_1^{m+1}, z_3^m)$. After replacing z_2^m by z_2^{m+1}, the same process is carried out for z_3^m, conditioned on the new values of z_1^{m+1}, z_2^{m+1}. Hence, each variable is updated conditioned on all other variables. In the update of the next variable, the current is replaced by its updated value.

Now, we can put this procedure in the context of SCSS for approximate likelihood calculation. Therefore, we are interested in the MAP estimate for each source. We can use Eq. (3.5) to formulate the GS method as described in algorithm 2:

```
Input: $\mathbf{Y}$, VQ bases for speaker 1 and 2
Output: $z_1^\star$, $z_2^\star$
Initialize: $m = 1$
Randomly initialize: $\{z_i^m\}$, $i \in \{1,2\}$, $j \in \big(\{1,2\} \setminus i\ \big)$
for $m \leftarrow 1$ to $M$ do
    $z_i^{\star,m+1} = \ \sim p(z_i^m|\mathbf{Y}, z_j^{\star,m})$
    $z_j^{\star,m+1} = \ \sim p(z_j^m|\mathbf{Y}, z_i^{\star,m+1})$
end for
```

Algorithm 2: Gibbs Sampling (GS).

Here, we only deal with the inference problem by computing the posterior and not with the entire parameter estimation process carried out during model training. Thus, the maximization step, usually performed in the EM agorithm for the parameter update, is not executed. This results in poor-convergence for the Gibbs Sampler. To overcome this problem, we introduce the iterated conditional modes algorithm, which is a greedy approximation of the Gibbs sampler. For the sake of completeness we nevertheless report results for the GS method in the experimental section 4.7.

Iterated conditional modes method: The iterated conditional modes algorithm (ICM) has been originally proposed for Markov random fields [112]. It is a time-consuming process to optimize the joint probability of the MAP estimate $p(z_1, z_2|\mathbf{Y})$ in Eq. (3.5). Therefore, in the ICM algorithm a *greedy* strategy is used to sequentially optimize the joint probability for one selected variable, while the remaining variables are kept constant, i.e. we iteratively optimize $z_i^\star = \text{argmax}_{z_i}\big[p(z_i|\mathbf{Y}, z_j^\star)\big]$ for $i \neq j$ and $i, j \in \{1,2\}$. The algorithm to optimize the MAP estimate for each time step via ICM is provided in Algorithm 3.

```
Input: $\mathbf{Y}$, VQ bases for speaker 1 and 2
Output: $z_1^\star$, $z_2^\star$
Initialize: $m = 1$
Randomly initialize: $\{z_i^m\}$, $i \in \{1,2\}$, $j \in \big(\{1,2\} \setminus i\ \big)$
for $m \leftarrow 1$ to $M$ do
    $z_i^{\star,m+1} = \text{argmax}_{z_i}\big[p(z_i^m|\mathbf{Y}, z_j^{\star,m})\big]$
    $z_j^{\star,m+1} = \text{argmax}_{z_j}\big[p(z_j^m|\mathbf{Y}, z_i^{\star,m+1})\big]$
end for
```

Algorithm 3: Iterated conditional modes algorithm (ICM).

Alternatively, we initialize the ICM algorithm four times randomly. The random variable associated to the first speaker is initialized twice as well as the second random variable, which corresponds to the second speaker. Finally, the best $\{z_1, z_2\}$ is selected among all four executed runs. We abbreviate this method by ICM4. Figure 4.1 illustrates how the ICM4 algorithm approaches the MAP solution for a single observation of the speech mixture. Each random initialization of the algorithm is labeled with a different color. The numbers denote the search path taken to reach the maximum in the observation likelihood plane. For this particular observation, the maximum is found by each of the random initializations.

Again, a naive implementation would have a complexity of $\mathcal{O}(TQ^2)$. Using the Gibbs Sampler or the ICM method as approximations of the likelihood calculation and assuming $M = 3$ iterations are sufficient to let the algorithm converge, i.e. at least to a local maximum,

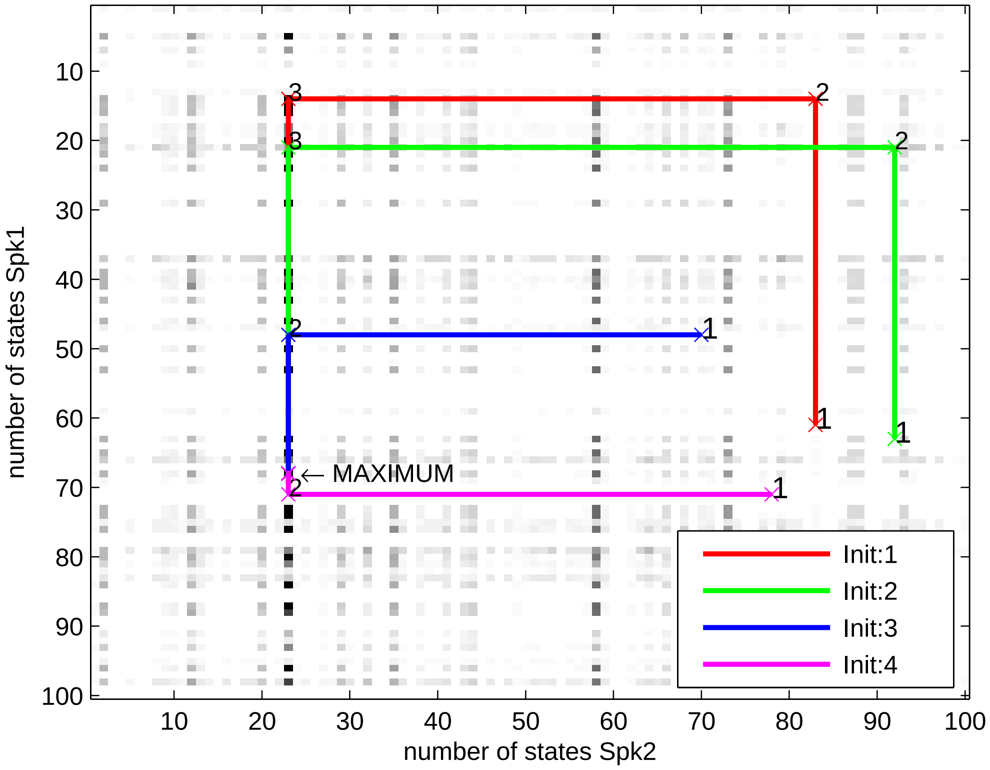


Figure 4.1: Observation likelihood plane for a single frame of a female-female speech mixture. The colored lines correspond to the four initializations of the ICM4 algorithm and the numbers denote the search path sequence taken to approach the maximum.

the computational complexity can be reduced to $\mathcal{O}(T \cdot M \cdot 2 \cdot Q)$. Note, the GS based algorithms are the first introduced approximations, which entirely split the posterior likelihood computation of a factorial model.

Finally, the Gibbs sampler and the ICM methods can also be applied to F-HMMs. In a factorial HMM, the Viterbi algorithm has to track the best path in the high dimensional space, i.e. one additional dimension for every hidden chain. Thus, the space would be quadratic, for a factorial model with two hidden chains, as assumed in the SCSS problem. Gibbs based techniques split this relation and either the transition probabilities can be directly integrated in the ICM algorithm as:

$$z_i^{\star,(\tau)} = \underset{z_i}{\operatorname{argmax}}\, p(z_i^{(\tau)}|z_i^{\star,(\tau-1)})\; p(z_i^{(\tau)}|\mathbf{Y}, z_j^{\star,(\tau)}), \tag{4.6}$$

where $z_i^{\star,(\tau-1)}$ denotes the best state for speaker i at time $\tau - 1$. For this extended model the computational complexity remains the same.

Furthermore, two independent instances of Viterbi decoders for each chain operating on the same observations can be applied:

$$w(z_i^{(\tau)}) = \log\; p\big(\mathbf{Y}^{(\tau)}|z_i^{(\tau)}, z_j^{\star,(\tau)}\big) \;+ \max_{z_i^{(\tau-1)}} \big(\log\; p(z_i^{(\tau)}|z_i^{(\tau-1)}) + w(z_i^{(\tau-1)})\big). \tag{4.7}$$

The extension to the framework of F-HMMs results in a reduced complexity of $\mathcal{O}(TM \cdot 8Q) + \mathcal{O}(T \cdot 2Q^2)$, where the first part accounts for the observation likelihood computation of the ICM4 and the second for tracking through the factorial model.

4.6 Computational Complexity

This section summarizes the computational complexity of the previously introduced suboptimal search heuristics. For the experiments, we use a vector quantizer (VQ) as statistical model to capture speaker dependent characteristics. Each SD VQ has cardinality $Q = |Z| = 500$. Hence, the training data was quantized into 500 cells. Table 4.1 summarizes the computational complexity of the introduced suboptimal search heuristics. The average length of the speech mixtures is 1.69 sec. Moreover, table 4.1 shows the average run time (AT) for the computation of the speech mixture separation of each algorithm. Additionally, we provide a real-time (RT) factor, which is simply given as AT dived by the average speech mixture length. Hence, the best performing algorithms with respect to the TMR, i.e. FBS and ICM4, require ~ 2.5 and ~ 4 times of the speech mixture length for separation. We are confident to improve the RT factor for both algorithms below one by efficient implementation. All experiments have been performed using MATLAB on an Intel CPU CORE-i7 QUAD 920 running at 2.66GHz.

Table 4.1: Complexity comparison for FM-VQ using full search (FS), fast likelihood estimation (FLE), fast beam search (FBS), Gibbs sampling (GS), and ICM4.

Method	Comp. Complexity	♯ of Evaluations	AT	RT Factor
FS	$\mathcal{O}(T\ Q^2)$	2.62e7	306.5	181.35
FLE	$\mathcal{O}(T\ (k^2\ + K^2))$	1.058e5	0.55	0.33
FBS	$\mathcal{O}(Q^2 + (T-1)\ N^2)$	5.096e5	4.4	2.6
GS	$\mathcal{O}(T\ M\ 2 \cdot Q)$	3.072e5	1.59	0.94
ICM4	$\mathcal{O}(4\ T\ M\ 2 \cdot Q)$	1.29e6	6.50	3.85

For the determination of the number of observation likelihood evaluations in Table 4.1, we assume to have $T = 100$ speech frames, which corresponds to 1 second of speech for a frame rate of 10 ms. We assume $k = K = 23$ for the fast likelihood estimation (FLE), as discussed in subsection 4.3. Moreover, for the FBS introduced in section 4.4 we set N equal 50. Finally, we assume that the number of iterations is $M = 3$ for the Gibbs sampling and ICM technique (see section 4.5). The complexity for all suboptimal heuristics can be reduced by approximately two orders of magnitude.

4.6.1 Performance and Cost Function

The observation likelihood in a factorial model determines the model quality of a specific state combination given the speech mixture as observation. Thus, the observation model includes a cost function, which measures the similarity between observed and hidden random variables. To evaluate the observation likelihood approximations, we measure how close these methods converge to the optimum, determined by the full search. To this end, the maximum of the observation log-likelihood for every speech segment of an utterance is determined. Afterwards, the time average for all test utterances is computed and the mean maximum likelihood is found by averaging over the whole test database. The mean optimum likelihood of each heuristics is marked on the average observation log-likelihood curve determined using FS. Figure 4.2 and 4.3 show the performance of the methods on a linear and logarithmic scale, respectively. The y-axis depicts the achieved log-likelihood of a specific state combination, whereas on the x-axis the state combinations are displayed in descending sorted log-likelihoods. Since each model has cardinality $Q = 500$, the total number of state combinations is 250000. Inspecting Figure 4.2, it seems that all approximations come quite close to the average optimum of the log-likelihood function achieved by the FS. In any case, all approximations find relevant state combinations, i.e. a state combination after the sharp bend of the curve leading to the maximum.

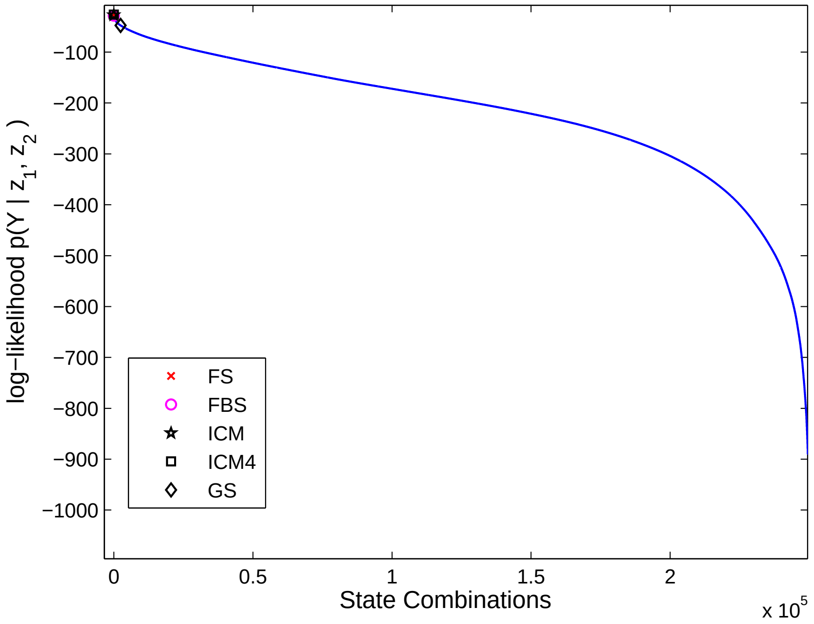


Figure 4.2: Mean log-likelihood function for all state combinations averaged over all test utterances. Log-likelihoods are sorted in descending order (linear scale). The markers identify the average score of the search heuristics.

The same plot on a log-log scale is shown in Figure 4.3. The maximum, achieved by the full search (FS) is marked by a red cross. Furthermore, the maximum has been found by the iterated condition modes algorithm with four initializations (ICM4), depicted as black square. Reasonable results are found in descending order by the ICM and the fast beam search (FBS). As already noted in section 4.5, the Gibbs sampler (diamond black marker) has poor convergence and thus, achieves the worst result.

4.7 Experiments and Results

In the previous section, the performance of the search heuristics has been measured and compared to the optimal log-likelihood cost function achieved by the FS. This section presents performance results in target-to-masker ratio (TMR) for source separation and studies the decrease in TMR for the approximation algorithms and compares them to the results of Figure 4.2.

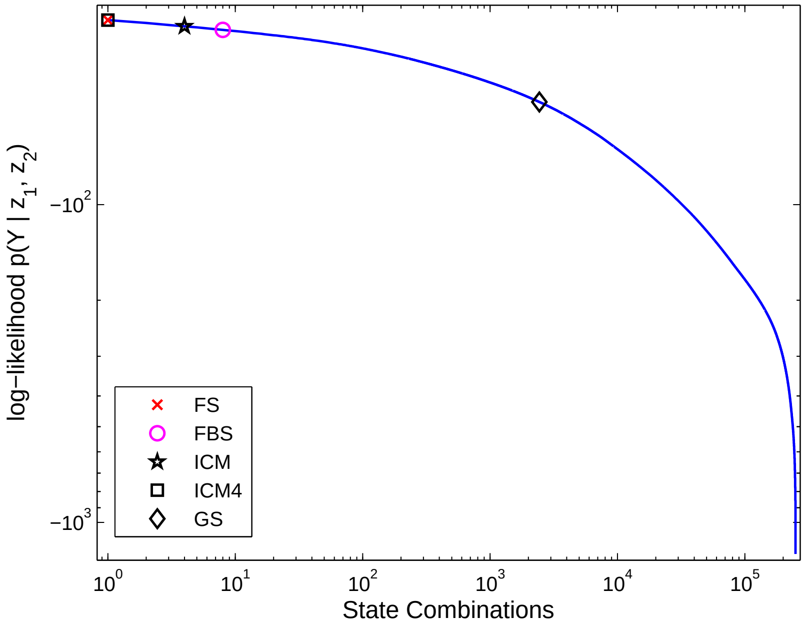


Figure 4.3: Mean log-likelihood function for all state combinations averaged over all test utterances. Log-likelihoods are sorted in descending order (logarithmic-scale). The markers identify the average score of the search heuristics.

For testing, all files are mixed at a level of 0 dB TMR and all possible combinations of target speakers and their interfering speakers are evaluated, resulting in altogether 54 mixed signals. Hence, 108 separated component signals are used for evaluation. To assess performance, all mixtures are split into the three following cases: (i) the same gender female (SGF), (ii) the same gender male (SGM), and (iii) the different gender (DG) cases. A detailed description of the experimental setup can be found in chapter 1, section 1.5. A comparison of the basic observation likelihood approximation methods in terms of TMR with mean and standard deviation can be found in Table 4.2.

Table 4.2: Separation results in TMR [dB] for the full search (FS), fast likelihood estimation (FLE), fast beam search (FBS), and iterated conditional modes (ICM) methods. Average TMR results with standard deviation (Std) are listed separately for the three different mixing cases.

Method		SGF	SGM	DG
FS	Mean	11.27	5.64	10.29
	Std	2.40	0.64	2.03
FLE	Mean	9.26	4.23	8.05
	Std	2.66	1.25	2.16
FBS	Mean	9.59	4.71	9.02
	Std	2.90	0.81	1.94
ICM	Mean	8.45	4.55	8.36
	Std	2.26	0.88	1.70

For the given model size Q, the results of the full search (FS) are the upper bound for all three cases, i.e. SGF, SGM, and DG. We note that none of the basic approximation techniques obtain the TMR of the FS. However, each of these methods can increase the TMR substantially. Moreover, the proposed FBS for all three cases has a superior performance compared to FLE and ICM for the specified setting. Separate experiments for the GS based methods and the FBS algorithm have been performed. For the GS based methods, first the GS algorithm is implemented. Second, we replace the sampling step of the GS by the maximum operator known as ICM algorithm. We employ performance for the ICM for two different settings: (i) We randomly initialize the ICM at every time step abbreviated by ICM. (ii) The ICM4 is four times randomly initialized (see Section 4.5). A comparison of all Gibbs based methods can be found in Table 4.3. Apparently, the GS method does not converge. Moreover, the ICM4 leads to a good approximation compared to the FS. This is also in accordance with the obtained likelihood in the cost function of Figure 4.3.

In Figure 4.4, generalized FBS results for various setups are compared to each other, as well as to the full search. The numbers in the label represent M and N (*FBS-M-N*), where M is the number of best states at time step $\tau - 1$, used to determine the N best states, which are actually used for decoding at time step τ. The results show that for a fixed value of $M = 1$, a larger value of N increases the TMR, but decreases the performance in terms of computational complexity. Increasing N to Q results in the FS method. However, fixing $N = 50$ and varying M from 1 to 3 leads to an increase of the TMR but keeps the computational complexity almost constant. A lager value of M just increases the computational burden during the selection of the best states and not the computational complexity for the factorial observation likelihood computation.

Figure 4.5 summarizes the results of the observation likelihood approximations. Specifically, the initial approximations (FBS, ICM) and their improved versions *FBS-3-50* and *ICM4* are compared. For convenience, the results of the full search (FS) and the FLE method are plotted as well. The proposed search heuristic *ICM4* approximates the MAP solution determined by full search best.

Table 4.3: Separation results in TMR [dB] for likelihood estimation, using different Markov chain Monte Carlo methods. Average TMR results with standard deviation (Std) are listed separately for the three different mixing cases.

Method		SGF	SGM	DG
Gibbs	Mean	4.92	4.16	5.76
	Std	1.10	1.29	1.50
ICM	Mean	8.45	4.55	8.36
	Std	2.26	0.88	1.70
ICM4	Mean	11.25	5.46	10.11
	Std	2.41	0.73	2.11

Finally, we apply the introduced likelihood approximations to factorial models with dynamics, the FM-HMM. The results are consistent with the FM-VQ results. A summary is illustrated in Figure 4.6. In this experiment, the beam width of the beam search was set to 500 (BS-500). We did not change the setting for all other suboptimal search heuristics. In order to perform tracking, the Viterbi algorithm is employed for all but the *max-ICM4* method. This method is the natural extension of the ICM4 to models with dynamics as introduced in Eq. (4.6). We emphasize that only the ICM based methods split the observation likelihood computation entirely. Moreover, the ICM4 approximation shows superior performance compared to all other discussed approximation heuristics. Note, the additional use of

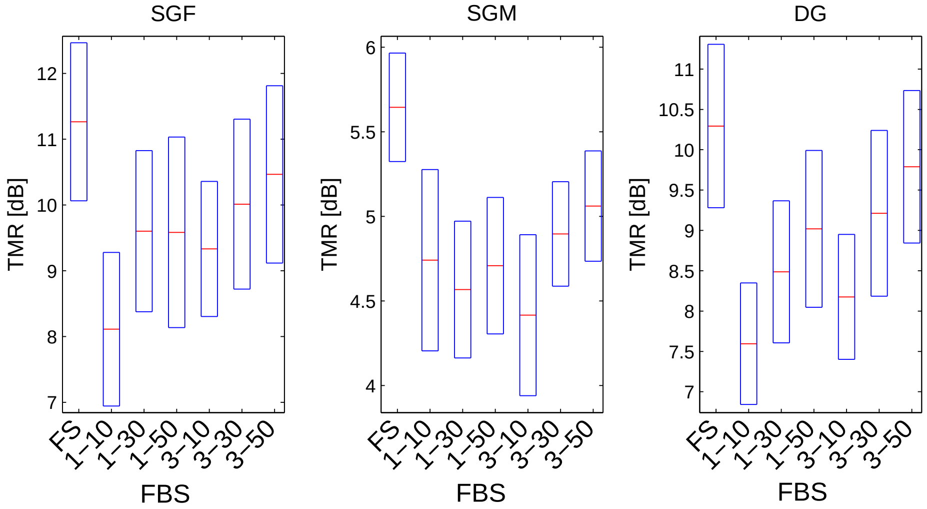


Figure 4.4: TMR results for various setups of the fast beam search (FBS) approximation. For decoding, the factorial-max VQ structure has been employed. The first number in the labels indicates the M-best states at $\tau - 1$ to determine the N states (second number) employed for separation at time step τ. See section 4.4 for more details.

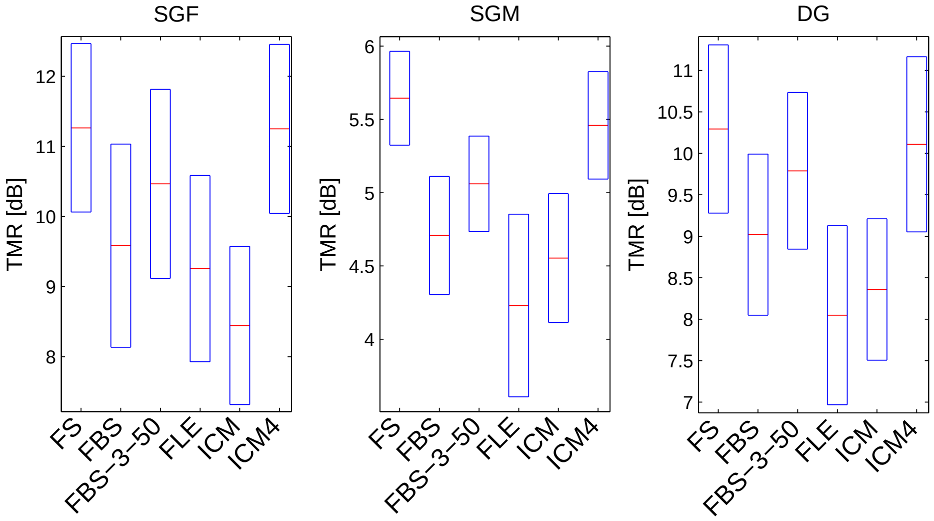


Figure 4.5: TMR result for factorial-max VQ observation likelihood approximations. Comparison of the baseline approximations (FBS, ICM) to their improvements (FBS-3-50, ICM4), as well as to the FS and FLE.

dynamics increases computational complexity but does not increase the TMR significantly.

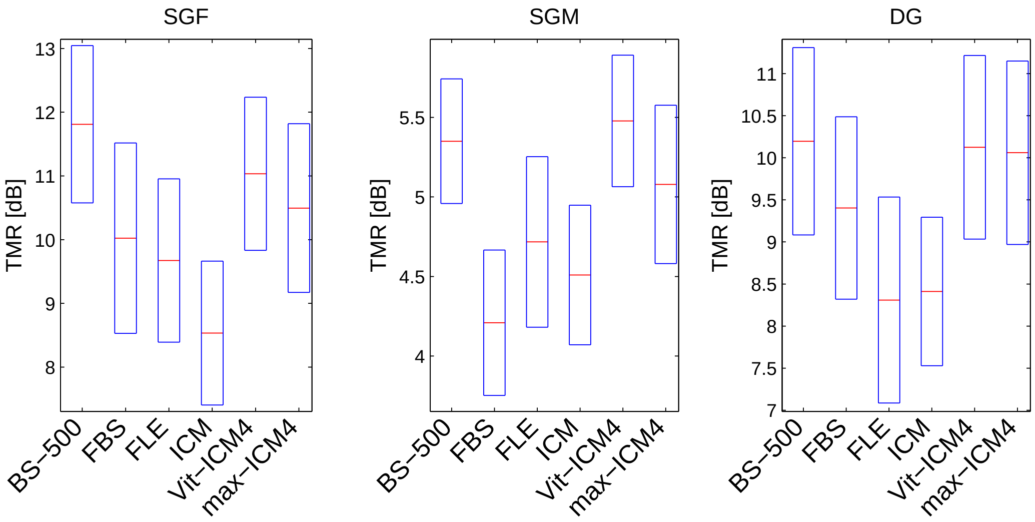


Figure 4.6: FM-HMM results using the approximation methods: beam search (BS), fast beam search (FBS), fast likelihood estimation (FLE), iterated conditional modes using Viterbi (ICM), ICM4 with Viterbi tracking (Vit-ICM4), and ICM4 with tracking (max-ICM4). The TMR is depicted separately for the SGF, SGM, and DG cases. See section 1.5 for more details.

4.8 Conclusion

This chapter introduced two new techniques to approximate the observation likelihood for models of factorial nature in order to reduce the computational complexity. These methods have been compared to the full posterior likelihood calculation and to the fast likelihood computation method, proposed in [108, 36]. We have shown that the complexity reduction for both methods is significant and results only in a slight decrease of performance in terms of TMR, compared to the full search. Additionally, we extended the heuristics for first order Markov processes, i.e. for F-HMMs. For the iterative conditional modes algorithm with re-initialization (ICM4), we can report no significant decrease in performance at all. Interestingly, there is strong evidence that the ICM4 method can be applied for factorial models with independent hidden variables. As a result, the corresponding Markov chains evolve independently over time. Additionally, we provided a real time factor, defined as the ratio of mean execution time to mean speech mixture length, for all algorithms. This factor shows that the ICM4 and the FBS methods are competitive candidates for real-time SCSS.

Chapter 5

Gain And Shape Modeling For Source Separation

5.1 Introduction

In model based single channel source separation (SCSS) generative models are trained for each speaker, using single speaker utterances. These utterances are usually normalized to their maximum amplitude in the time domain. Afterwards, the signals are transformed to a time-frequency representation. Finally, the trained models are employed for source separation. In this process, the codewords of a VQ or the mean values of a GMM/HMM are employed as emission density means. These mean values are utilized as prototypes to represent the given speech mixture observation. Throughout this chapter we employ the *mixmax* interaction model of chapter 3, Eq. (3.16) as model combination operator. Following this procedure, each speaker dependent (SD) model is trained on the same energy level. Hence, the models are trained to perform best for component signals mixed at equal level. For other mixing levels however, there is a mismatch for these trained models.

Recently, Kristjansson et al. [5] proposed to estimate the mixing level measured in the signal-to-noise ratio (SNR), which is similarly defined as the TMR in Eq. (1.2), for the whole speech utterance *a priori*. For this purpose, speech frames originating from a single source are identified. In their work, speaker identity and mixing level are estimated simultaneously on just these frames. This is done in an expectation-maximization like fashion. In the expectation step, the likelihood for a hypothesized speaker pair is calculated, whereas in the maximization step the appropriate mixing level is estimated. Before separation, the identified speaker dependent models are globally adjusted by the estimated TMR. Therefore, the whole utterance must be available in advance. Hence, this method cannot be applied for online separation. Additionally, in [5] the TMR is selected out of a discrete finite set, which also seems to be restrictive.

In Radfar et al. [24] the mixing level has been estimated in a maximum likelihood based way. Here, the state combinations have been maximized for each mixing level taken from a discrete set. The state combinations and mixing levels with the highest likelihood, averaged over the whole utterance are finally employed for separation. Thus, this procedure results in an M-times higher complexity, where M is the number of different discrete mixing levels.

Non-negative matrix factorization (NMF) [65] decomposes the observation into a weight and bases matrix. The contribution of each basis stored in a dictionary is thus estimated

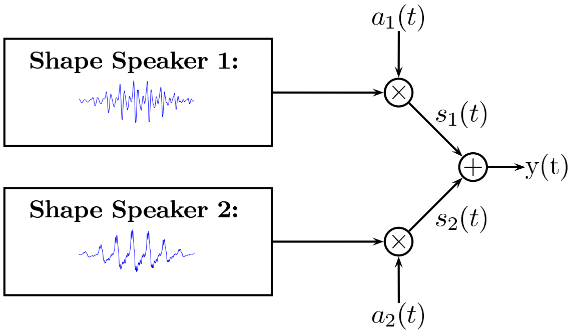


Figure 5.1: Shape-Gain production model

inherently in the algorithm. NMF estimates the weights on a frame basis and assigns to each basis a weight equal or greater than zero. Thus, the mixture is approximated by a weighted combination of all SD bases, which is in spirit akin to the posterior mean. In this chapter however, the explicit gain estimation for a particular state is of interest. Therefore, NMF will be adapted in section 5.2.5.

In contrast to previous work, we propose to estimate the gain associated to each speaker for every speech segment separately. This gain estimation replaces the estimation of the mixing level. It provides the benefit to be applicable for online processing without restriction to a fixed discrete set. This is motivated by the shape-gain Vector Quantizer (VQ), proposed by Sabin [50] to encode single speech utterances. The shape-gain VQ decomposes the coding problem into that of coding a scalar, the gain, and a vector, the shape. The separate shape and gain coding is motivated by the fact that in speech the same shapes can be produced at different levels. Vector quantizers, however, model shapes of different gains with separate codewords.

In contrast to the shape-gain VQ, the gain is not selected from a discrete set, a codebook, but instead is estimated during decoding. Hence, the gain is an online determined continuous valued estimate. The underlying shape-gain production model for source separation is shown in Figure 5.1. In the model, each speakers' normalized shape of a time frame is weighted by the gain factor a_i. The mixture $y(t)$ is the summation of the frames of speaker $s_1(t)$ and $s_2(t)$. Note that shapes correspond to normalized codewords or state means. In the sequel, we will use this denomination interchangeably.

Problem formulation: The gain estimation for source separation can be seen as an optimization problem. In particular, it is a linear least squares problem with linear inequality constraints:

$$\begin{aligned} \text{minimize:} \quad & \min_{x} \ f(x) \\ \text{subject to:} \quad & g(x) \geq h \end{aligned} \tag{5.1}$$

Here, the objective function $f(x)$ is the quantity to be minimized and $g(x)$ are the inequality constraints. To solve linear least squares problem with linear equality constraints, the method of Lagrange multipliers [106] and [113, Ch. 5] can be used. Problems with linear inequality constraints however have to be solved using the Karush-Kuhn-Tucker Condition [114].

In particular, for a given observation $\mathbf{y}$ and given normalized prototype shapes $\overline{\mathbf{s}}_1$ and $\overline{\mathbf{s}}_2$ of the hidden variables z_1 and z_2 the optimization problem can be formulated as follows:

$$\begin{aligned} \text{minimize:} \quad & \min_{a_1,a_2} \ || \ \overline{S}\mathbf{a} - \mathbf{y} \ ||_2 \\ \text{subject to:} \quad & a_1, a_2 \geq 0, \end{aligned} \tag{5.2}$$

where $\overline{S} = \left[\overline{\mathbf{s}_1},\ \overline{\mathbf{s}_2}\right]$ and $\mathbf{a} = [a_1,\ a_2]^T$ are the normalized speech shapes and the unknown gain variables, respectively. Assuming independent shapes, each gain variable can be optimized separately. This case is called *singular variable optimization.* The parallel optimization of the gain variables is know as *multi-variable optimization.*

In the sequel, different methods are introduced to solve the shape-gain problem as formulated in Eq. (5.2). Therefore, we split the problem and separately address the problem of gain estimation in section 5.2 and shape estimation in section 5.3. In the experiments, gain estimation performance will be assessed for known shapes. Finally, we combine the gain and shape estimation and perform SCSS.

5.2 Gain Estimation

5.2.1 Maximum-Likelihood Gain Estimation

In this section, the gain is modeled as the weighting a^k of the Gaussian mean μ^k of a Gaussian mixture model (GMM) with $k \in \{1, \ldots, K\}$ components. Here, we assume that the observed single speaker data $\mathbf{s}$ follow a Gaussian distribution:

$$p(\mathbf{s}|\mu, a) = \mathcal{N}(\mathbf{s}; a \cdot \mu, \sigma^2),$$

where $\mathcal{N}$ is the normal distribution, $a \cdot \mu$ is the gain adjusted density mean, and σ^2 is the variance. This gain model is similar in spirit to that proposed by Bimbot et al. [84]. In contrast to their work, we do not assume Gaussians with zero mean and do not derive the gains for the variance of the Gaussians in a GMM. The definition of the gain-shape GMM for single speaker speech is as follows:

$$p(\mathbf{s}|a^1, \ldots, a^K) = \sum_{k=1}^{K} \omega^k \frac{1}{\sqrt{2\pi}\ \sigma} \cdot \exp - \frac{\left(\mathbf{s} - a^k\ \mu^k\right)^2}{2\sigma^2}, \tag{5.3}$$

where $\mathbf{s}$ is the spectrum of the observed speech frame, μ^k is the Gaussian mean value of component k, and a^k are the gains associated to each Gaussian component. Note, in this model the Gaussian means represent the shapes. Further, ω^k is the weighting of each Gaussian, where ω^k fullfills the constraint to be positive $\omega^k \geq 0$ and $\sum_k \omega_k = 1$. σ^2 is assumed to be the diagonal covariance. The model of Eq. (5.3) can be extended to that of observing a mixture of two speakers:

$$p(\mathbf{y}|\ k_1, k_2, a_1^{k_1}, a_2^{k_2}) = \frac{1}{\sqrt{2\pi}\ \sigma} \cdot \exp - \frac{\left(\mathbf{y} - (a_1^{k_1}\mu_1^{k_1} + a_2^{k_2}\mu_2^{k_2})\right)^2}{2\sigma^2}. \tag{5.4}$$

Note, For a given observation and given shapes $\mu_1^{k_1}$ and $\mu_2^{k_2}$, the goal is the maximization of the objective function subject to a_1 and a_2 under the positivity constraint:

$$\{\hat{a}_1^{k_1}, \hat{a}_2^{k_2}\} = \underset{a_1^{k_1} \geq 0,\ a_2^{k_2} \geq 0}{\operatorname{argmax}}\ p(\mathbf{y}|\ k_1, k_2, a_1^{k_1}, a_2^{k_2}).$$

The quality of the estimation of the parameters depends on the variance of the assumed density function. Thus, the "sharpness" of the Gaussian likelihood function determines the estimation accuracy of the unknown parameters [115]. The variance of parameters in the likelihood function is measured by taking the negative of the logarithm of the objective function $\gamma(\mathbf{y})$ and differentiate twice, i.e., $-\frac{\partial^2 \log\gamma(\mathbf{y})}{\partial \mathbf{a}^2}$. For the ML-based gain estimation however, we take the logarithm of the likelihood function $p(\mathbf{y}|\ k_1, k_2, a_1^{k_1}, a_2^{k_2})$ as:

$$\log\gamma(\mathbf{y}) = -\log(\sqrt{2\pi}\ \sigma) - \frac{1}{2\sigma^2}\big(\mathbf{y} - (a_1\mu_1 + a_2\mu_2)\big)^2, \quad i \in \{1,2\}. \tag{5.5}$$

Subsequently, the dependency on the Gaussian component k of the GMM will be omitted, for simplicity. To optimize Eq. (5.5) with respect to a_i under the constraint that $a_i \geq 0$ we can define the Kuhn-Tucker equations:

$$\begin{aligned} -\log(\sqrt{2\pi}\ \sigma) - \frac{1}{2\sigma^2}\big(\mathbf{y} - (a_i\mu_i + a_j\mu_j)\big)^2 + \lambda_i(a_i - c) &= 0 && (5.6)\\ \lambda_i(a_i - c) &= 0 && (5.7)\\ \lambda_i &\geq 0, \end{aligned}$$

where we set $c = 0$.

Differentiating Eq. (5.6) with respect to a_i, i.e. $\frac{\partial \log\gamma(\mathbf{y})}{\partial a_i}$ and combining it with Eq. (5.7) finally results in:

$$a_i\mu_i \cdot \frac{\mathbf{y} - (\ a_i\mu_i + a_j\mu_j\ )}{\sigma^2} = 0, \tag{5.8}$$

where $i \neq j$ and $i, j \in \{1,2\}$. Now, we can find an iterative additive update procedure, by using a gradient descent method as described in [116]:

$$a_i^{(l+1)} = a_i^{(l)} + \zeta\ \Big[\frac{\mu_i\mathbf{y}}{\sigma^2} - \frac{\mu_i}{\sigma^2}\sum_{j=1}^{2} a_j^{(l)}\mu_j\Big], \quad i \in \{1,2\}, j \in (\{1,2\} \setminus i) \tag{5.9}$$

where ζ is in general a small constant greater than zero and l denotes the iteration. Thus, ζ controls the impact of changes on $\mathbf{a}$ from iteration l to iteration $l+1$. To move from the additive update scheme to a multiplicative update [66], we set $\zeta = a_i\frac{\mu_i}{\sigma_i^2}\sum_{j=1}^{2} a_j\mu_j$, which results in:

$$a_i^{(l+1)} = a_i^{(l)}\ \sum_{d=1}^{D}\Big(\frac{\mu_i(d)\ \mathbf{y}(d)}{\sigma^2(d)}\Big)\ /\ \sum_{d=1}^{D}\Big(\frac{\mu_i(d)\sum_{j=1}^{2} a_j^{(l)}\mu_j(d)}{\sigma^2(d)}\Big), \tag{5.10}$$

where $d = [1, \ldots, D]$ is the frequency bin index. A simplification of the update rule can be achieved by replacing the variance $\sigma^2 = (\sum_{j=1}^{2} a_j\mu_j)^2$ in Eq. (5.10). The new update formula of Eq. (5.11) is akin to that proposed in [84] for variance gain estimation:

$$a_i^{(l+1)} = a_i^{(l)}\ \sum_{d=1}^{D}\Big(\ \frac{\mu_i(d)\ \mathbf{y}(d)}{\sigma(d)^2}\Big)/\sum_{d=1}^{D}\Big(\frac{\mu_i(d)}{\sigma(d)}\Big), \tag{5.11}$$

The whole gain estimation method is summarized in algorithm 4, where in line 8 Eq. (5.10) can be replaced by its simplified version of Eq. (5.11). Note, the value of the variance is noncritical. Either a small constant or the true variances, which have to be adjusted accordingly at each iteration as $\sigma = (\sum_{j=1}^{2} a_j\sigma_j)$, can be used. Furthermore, note that, for the multiplicative update scheme, no parameters have to be tuned.

The multi-variable ML-based estimation can be similarly applied for singular variable estimation.

```
input: $\mathbf{y}(d)$, VQ bases $\mu_1, \mu_2$ for speaker 1 and 2
output: $a_1$, $a_2$
randomly initialize: $a_1^{(1)}$, $a_2^{(1)}$
initialize: $l = 1$, $g^{(0)} \gg$, and $\epsilon \ll$
repeat
    $g^{(l)}(d) = a_1^{(l)} \mu_1(d) + a_2^{(l)}\ \mu_2(d)$
    for $i = 1$ to $2$ do
        $a_i^{(l+1)} = a_i^{(l)}\ \sum_d \left(\frac{\mu_i(d)\mathbf{y}(d)}{\sigma^2(d)}\right)\ /\ \sum_d \left(\frac{\mu_i(d) g^{(l)}(d)}{\sigma^2(d)}\right)$
    end for
    $l = l + 1$
until $(a_1^{(l+1)} - a_1^{(l)} < \epsilon)\ \cap\ (a_2^{(l+1)} - a_2^{(l)} < \epsilon)$
```

Algorithm 4: Iterative multi-variable ML-based gain estimation.

5.2.2 Nonlinear Gain Estimation

Alternatively, the gain factor can be estimated as an additive component in the logarithmic domain. We employ a nonlinear method based on percentile filtering for estimation. In general, the gain normalized speech segment $\overline{\mathbf{S}}$ and the speech segment with gain $\mathbf{S}$ have the following relation in the log-domain:

$$\mathbf{S}_i = \mathbf{a}_i + \overline{\mathbf{S}_i} \rightarrow \mathbf{a}_i = \mathbf{S}_i - \overline{\mathbf{S}_i}, \tag{5.12}$$

where $\mathbf{a}(d)$ is the gain vector containing the same value for each vector entry, i.e. $\mathbf{a}(d) = \text{const.} \equiv a \cdot \mathbf{1}$, where $\mathbf{1}$ is a vector with all components equal to one. If the true normalized speech frames are however replaced by the density means μ_i, the representation does not match exactly anymore. This results in a gain vector containing different values in $\mathbf{a}$. In order to tackle this problem we have to estimate the gain for each speech segment. The Gaussian probability density model of Eq. (5.3) basically measures the similarity between the speech segment $\mathbf{Y}_i$ and $\mu_i^{k_i}$.

The probability density function for the speech mixture of Eq. (3.16) in the log-frequency domain is given as:

$$p(\mathbf{Y}|\ k_1, k_2) = \frac{1}{\sqrt{2\pi}\ \sigma} \exp\Big(- \frac{\big(\mathbf{Y} - \max(\mu_1^{k_1},\ \mu_2^{k_2})\big)^2}{2\sigma^2}\Big), \tag{5.13}$$

where we represent $\mathbf{S}_i$ by $\mu_i^{k_i}$. The gain factors a_i should be determined such that the likelihood of observing $\mathbf{Y}$ is maximized. We discovered that we can find each gain factor independently. In order to estimate the gain of a speakers' density mean given the observed speech mixture $\mathbf{Y}$, we adapt Eq. (5.12) to $\mathbf{a}_i(d) = \mathbf{Y} - \mu_i^{k_i}$ and perform percentile filtering [92] on the gain vector $\mathbf{a}_i(d)$. In contrast to the percentile filtering as defined in [92] where the filtering is performed over time, we define the percentile filtering over frequency. Therefore, the gain vector is first sorted in ascending order:

$$\mathbf{a}_i(\rho_0) \leq \mathbf{a}_i(\rho_1) \leq \ldots \leq \mathbf{a}_i(\rho_D). \tag{5.14}$$

The estimate for the gain is obtained by taking the r^{th}-percentile as $\hat{a} = \mathbf{a}_i(\lfloor rD \rfloor)$, where $0 \leq r \leq 1$ and $\lfloor \cdot \rfloor$ indicates the element-wise rounding operator. Taking the value $r = 0$ corresponds to the minimum in $\mathbf{a}$ and $r = 0.5$ to the median. For noise estimation, the median is considered to be a robust estimator.

5.2.3 Projection based Gain Estimation

In this section, we follow the notion of orthogonal matching pursuit (OMP) [117] [118] to estimate the gain factor of each speech basis. In OMP, dictionary atoms are estimated by projecting the observed signal in the space spanned by the dictionary. In general, we suppose that the dictionary is complete and may have an infinite number of atoms. An usual application of OMP is signal coding, where the C best matching atoms with their respective gain estimates are selected in order to approximate the signal. For example, a signal frame $\mathbf{s}(t)$ is represented by the weighted sum of dictionary atoms $h_c(t)$ plus a residual component $r(t)$ as:

$$\mathbf{s}(t) = \sum_{c=1}^{C} \mathbf{a}(c) h_c(t) + r(t) \tag{5.15}$$

Afterwards, the indices associated to the dictionary atoms and their gains are optionally transmitted and used to represent the signal. The orthogonality property of the dictionary atoms enables the iterative estimation of the dictionary elements. Here, elements are selected using the inner product, defined as $\langle s(t), h(t) \rangle = \sum_{t=1}^{T} \mathbf{s}(t)\ h(t)$. The gain associated to each atom is estimated by the same equation, given that each dictionary atom is normalized to unity gain.

For SCSS, the dictionaries are represented by speaker dependent (SD) trained statistical models, i.e., VQ codebooks. Moreover, the gain determination is carried out in the magnitude frequency domain. In contrast to OMP, the state means of an SD model are in general not orthogonal to each other. However, we assume independency or at least quasi-orthogonality between the models of two speakers. Roweis [69] has shown that this assumption is valid. Therefore, the gain factor of each SD speech shape can be determined independently of the respective other speaker. We employ OMP for gain estimation only and not to determine specific dictionary elements. The gain associated to a dictionary atom of a particular speaker is simply determined as: $a(z_i) = \langle \mathbf{y}, \mu(z_i) \rangle$. Thus, this idea is akin to shape-gain vector quantization as described in [119].

5.2.4 Nonnegative Least Squares based Gain Estimation

The nonnegative least squares algorithm [114] uses the Kuhn-Tucker condition to solve the problem, as defined in Eq. 5.1 in an iterative manner. At the beginning, the algorithm sets all entries of $\mathbf{a}$ to zero. Afterwards, the signal is projected into the space spanned by the SD models as already shown in the projection based algorithm (see section 5.2.3). The codeword for the VQ with the highest positive correlation is selected. For this given codeword the gain factor a_i is estimated using the least squares solution of $\mu_i \cdot a_i \simeq \mathbf{y}$. Thus, a_i is determined by

$$a_i = (\mu_i^T \mu_i)^{-1} \cdot \mu_i^T \mathbf{y}.$$

Only if this determined gain factor a_i is positive, the codeword is kept. The resulting weighted basis is subtracted from $\mathbf{y}$ and the residual is utilized for the projection, described at the beginning. These steps are carried out until a minimum error criterion is met or no codewords, which fullfill the above constraint, are found. In our case, only the gain estimation procedure is of interest. Thus, we constrain the algorithm to estimate either a single gain value for one input shape or alternatively two gain values for two input shapes.

5.2.5 Nonnegative Matrix Factorization based Gain Estimation

Nonnegative matrix factorization (NMF) has been already introduced in chapter 2, section 2.2.2. Originally, NMF was introduced to code parts of objects [65] by dividing observation in a weight and a bases matrix. In the context of shape-gain coding of speech mixtures for speech separation, we employ NMF just for gain estimation. Therefore, in the NMF update rules, only the update of the weights is used and the update for the bases, which correspond to the shapes, is disabled.

5.2.6 Auditory Motivated Gain Estimation

In this section, we utilize a property of the human auditory system to estimate the gain independently for each speaker. Specifically, the human auditory masking property in the frequency domain is employed [8]. Let us assume two different sound events played at the same time in the same frequency range at different sound levels. Masking refers to the property that the sound played at the lower level is inaudible because of the louder sound. Thus, the sound at lower level is masked by the sound at higher sound pressure level. This directly results in the motivation of the binary mask, discussed in chapter 3, section 3.4. In this section, the gain is also estimated as an additive component in the logarithmic frequency domain as done in section 5.2.2. Therefore, we employ the described masking property and assume that each time-frequency cell with high energy is occupied by just one speaker. This is in accordance with the sparseness assumption of speech in a high resolution time-frequency representation. The auditory gain estimation is illustrated in algorithm 5. (i) Select high energy frequency cells in each emission density mean μ_i of both speaker dependent models. This is done by calculating all critical points d_C, by setting the first derivative to zero, as shown in line four. Note, the prime operator $\mu_i'(d)$ indicates the first derivative. In line five, Fermat's Theorem [1] is employed to select all d_M local maxima from the critical points. (ii) From this set of frequency bins d_M, the subset $d_m \subset d_M$ with highest energy remains. Here, d_m comprises $|d_m| = M$ elements. Experimentally, a value of $M = 10$ was determined, due to it's good performance. (iii) Subsequently, the difference ϵ of the observation $\mathbf{Y}$ and μ_i at the selected indices of maxima is calculated. The gain a_i is finally determined by taking the median of ϵ. This algorithm has been adapted from that introduced in [120].

```
1: input: $\mathbf{Y}$, VQ basis $\mu_1, \mu_2$ for speaker 1 and 2
2: output: $\hat{a}_1$, $\hat{a}_2$
3: for $i = 1$ to 2 do
4:     Find critical points $d_C$: $d_c = \{d|\mu_i'(d) = 0\}$
5:     Find maxima of $d_C$: $d_M = \{d_c|\mu_i''(d_C) < 0\} \quad c \in \{1, \ldots, C\}$
6:     Sort levels of maxima in descending order: $\mathrm{sort}(\mu_i(d_M))$
7:     Select first N indices of $d_M$ with highest level: $d_m \subset d_M$, where $|d_m| = N$
8:     Compute: $\chi = \mathbf{Y}(d_m) - \mu(d_m)$
9:     Estimate gain $\hat{a}_i$: $\hat{a}_i = \mathrm{median}(\chi)$
10: end for
```

Algorithm 5: Auditory motivated gain estimation.

[1] This theorem gives instructions how to find local maxima and minima of differentiable functions.

5.3 Combined Shape and Gain Estimation

So far, only the estimation of the unknown gains for given shapes has been discussed. This section introduces three structures to estimate both, gain and shape. The first structure, denoted as shape-gain decoder, determines the shape first and the gain afterwards. Secondly, the gain-shape decoder estimates the gains of both speakers simultaneously. Subsequently, for the predetermined gains, the most likely shapes are estimated. Thirdly, for each SD shape, the gain is estimated independently. Following, the gains and shapes are used to construct prototype densities which are employed for separation.

5.3.1 Shape-Gain Decoder

The shape-gain decoder, as introduced in [50] determines in a first step the shape vector for a given observation. This is accomplished by projecting the observation in the unitary space of the shape vector codebook. The best matching shape of the shape vector codebook is further employed for gain estimation. Afterwards, for the given shape the best gain is selected from the gain codebook. In [50], the separate coding of shape and gain was applied to linear predictive coding (LPC) coefficients and speech waveforms. This method was designed for single-speaker speech only. An extension of this method in order to decode two speakers talking simultaneously, is shown in Figure 5.2. This decoder structure can be embedded naturally in the already introduced ICM decoding algorithm (see chapter 4, section 4.5). Here, speaker separation is carried out by sequential speaker decoding. At the beginning, the shape of the first speaker with best match is selected. Afterwards, the gains are estimated for the determined shape of the first speaker and a randomly selected shape for the second speaker. The estimation of the gain factors is indicated by the distance function, shown in the Figure 5.2. The function $d(a, b)$ determines the distance between a and b and the minimization of this function is carried out by the methods, discussed in the previous sections. From here, the estimated gain of the first speaker is used to calculate a residual signal $\mathbf{r}$. This residual is employed to determine the best matching shape of the second speaker. Finally, the gains are estimate for the given shapes. This procedure is iterated as described for the ICM4 algorithm in chapter 4, section 4.5. Note, this decoder structure can be used for both, multi- and singular-variable estimation.

5.3.2 Gain-Shape Decoder

The gain-shape decoder structure as shown in Figure 5.3 naturally fits in the maximum-likelihood framework of factorial models employed for SCSS. However, the computational complexity is also of factorial nature. In this structure, the gain is firstly estimated for all shape codeword combinations of two speakers. This is achieved by minimizing a certain distance function $d(a, b)$. Afterwards, in the shape estimation module the adjusted codewords are employed for MAP estimation. Note that this decoder structures always perform multi-variable gain optimization.

5.3.3 Independent Speaker Decoder

The simultaneous speech of two speakers in general can be assumed to be independent. Consequently, we can expect that also the gains of each speakers' shape are independent and hence can be estimated independently. Figure 5.4 depicts a structure, where the gain of each shape codeword of a speaker dependent VQ is estimated independently of the competing

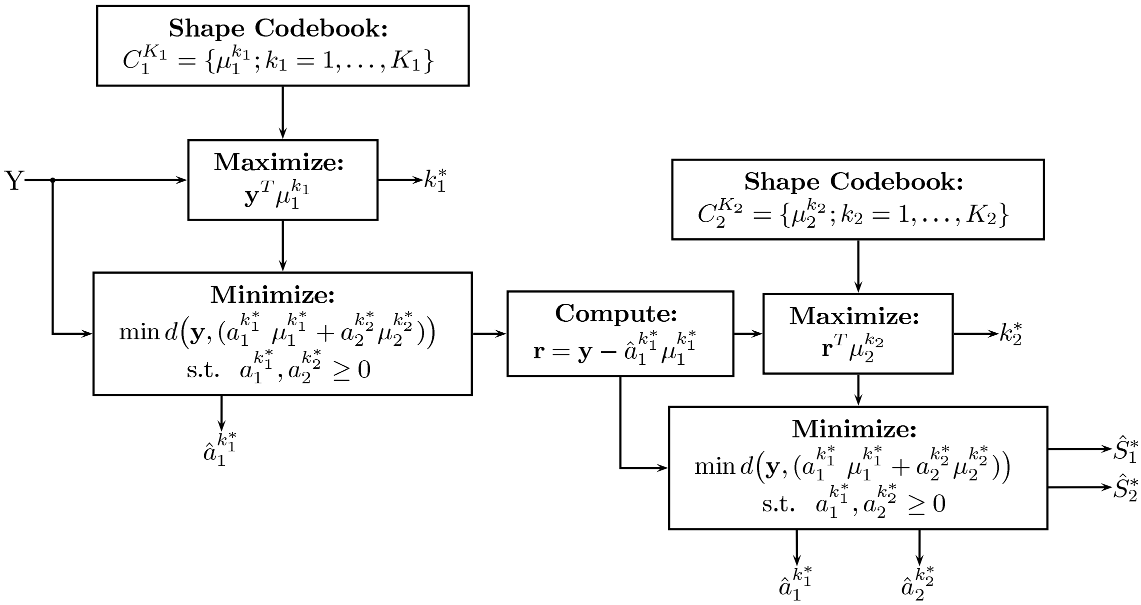

Shape Codebook:
$C_1^{K_1} = \{\mu_1^{k_1}; k_1 = 1,\ldots,K_1\}$
Y
Maximize:
$\mathbf{y}^T \mu_1^{k_1}$
k_1^*
Shape Codebook:
$C_2^{K_2} = \{\mu_2^{k_2}; k_2 = 1,\ldots,K_2\}$
Minimize:
$\min d(\mathbf{y}, (a_1^{k_1^*} \mu_1^{k_1^*} + a_2^{k_2^*} \mu_2^{k_2^*}))$
s.t. $a_1^{k_1^*}, a_2^{k_2^*} \geq 0$
Compute:
$\mathbf{r} = \mathbf{y} - \hat{a}_1^{k_1^*} \mu_1^{k_1^*}$
Maximize:
$\mathbf{r}^T \mu_2^{k_2}$
k_2^*
$\hat{a}_1^{k_1^*}$
Minimize:
$\min d(\mathbf{y}, (a_1^{k_1^*} \mu_1^{k_1^*} + a_2^{k_2^*} \mu_2^{k_2^*}))$
s.t. $a_1^{k_1^*}, a_2^{k_2^*} \geq 0$
$\hat{S}_1^*$
$\hat{S}_2^*$
$\hat{a}_1^{k_1^*}$
$\hat{a}_2^{k_2^*}$

Figure 5.2: Shape-gain decoder structure for the ICM4 based observation likelihood computation. The gain factor computational complexity is significantly reduced.

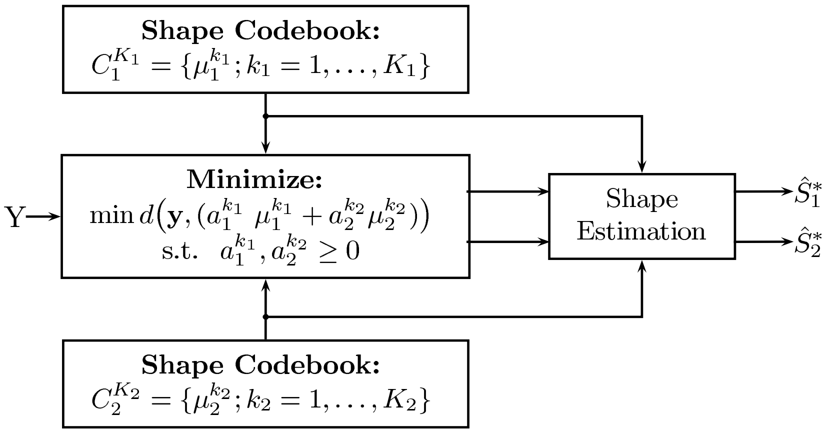

Shape Codebook:
$C_1^{K_1} = \{\mu_1^{k_1}; k_1 = 1,\ldots,K_1\}$
Y
Minimize:
$\min d(\mathbf{y}, (a_1^{k_1} \mu_1^{k_1} + a_2^{k_2} \mu_2^{k_2}))$
s.t. $a_1^{k_1}, a_2^{k_2} \geq 0$
Shape Estimation
$\hat{S}_1^*$
$\hat{S}_2^*$
Shape Codebook:
$C_2^{K_2} = \{\mu_2^{k_2}; k_2 = 1,\ldots,K_2\}$

Figure 5.3: Gain-shape decoder structure for multi-variable gain optimization methods.

speaker. Thus, for this structure only singular variable estimation methods can be employed. At every instant of time, the gain associated to a shape codeword of a speaker is determined. Once all gains are computed the most likely codeword combination can be identified. For this structure, all heuristics approximating the observation likelihood can be employed.

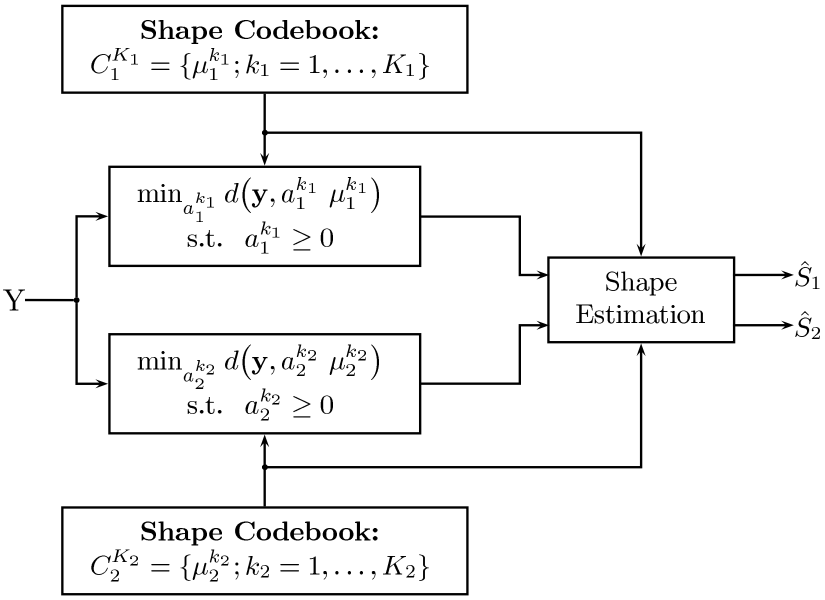

Shape Codebook:
$C_1^{K_1} = \{\mu_1^{k_1}; k_1 = 1, \ldots, K_1\}$
$\min_{a_1^{k_1}} d(\mathbf{y}, a_1^{k_1}\ \mu_1^{k_1})$
s.t. $a_1^{k_1} \geq 0$
Y
Shape Estimation
$\hat{S}_1$
$\hat{S}_2$
$\min_{a_2^{k_2}} d(\mathbf{y}, a_2^{k_2}\ \mu_2^{k_2})$
s.t. $a_2^{k_2} \geq 0$
Shape Codebook:
$C_2^{K_2} = \{\mu_2^{k_2}; k_2 = 1, \ldots, K_2\}$

Figure 5.4: Gain-shape decoder structure with independent gain estimation for each mean value associated to a hidden variable.

5.4 Experiments and Results

Performance of the six introduced gain estimation methods is assessed in two different experiments. Firstly, the gain estimation performance is evaluated exclusively. Therefore, the shapes are assumed to be a priori known and only the gains, observing the speech mixture, are estimated. The normalized speech frames of each speaker are employed as shapes. The second experiment investigates both, the gain and shape estimation. The decoding structures introduced in section 5.3 are used for this task.

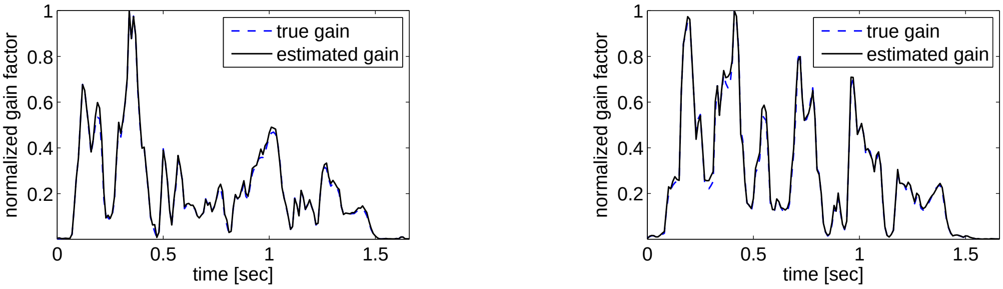
5.4.1 Supervised Gain Estimation

In this section we assess performance of all gain estimation methods. Therefore, two speech signals are mixed at TMR levels of 0, 3, 6, and 9 dB. Afterwards, all signals are transformed to the frequency domain and each component speech signal, i.e. S_1 and S_2, is normalized. For the methods operating in the log-frequency domain, the signal segments are normalized such that the maximum frequency component has 0 dB. Signals in the magnitude frequency domain are normalized to unit norm. These normalized component speech segments are abbreviated as $\overline{\mathbf{s}}_i$ and $\overline{\mathbf{S}}_i$ in the magnitude or log-magnitude domain, respectively. In this experiment, the gain estimation capability of each method is investigated. Therefore, the impact of shape estimation errors is excluded, using the normalized component speech segments as reference shapes. This results in the gain estimation for every single speech segment for an observed speech mixture and given reference shapes. Finally, the normalized component signal segments are weighted by the gain estimates and compared to the true signals.

Figures 5.5,5.6, 5.7, 5.8, 5.9, and 5.10 (a) and (b), respectively, compare the gain estimates (black solid line) to the true gains (blue dashed lines) for two female component speakers over time. For these experiments, the underlying signals are mixed at equal level. The amplitudes are normalized to the range between zero and one in the plots. For evaluation, the gain estimates are found for given normalized speech segments $\overline{S}_i$ and the observed speech mixture Y. From visual inspection, we observe that the gain estimates of all methods can follow the true gains quite well.

For all gain estimation methods, the TMR results for both, the target (t) and the masker (m) speech signal are depicted in table (c) of Figure 5.5, 5.6, 5.7, 5.8, 5.9, and 5.10, respectively. Results are shown for different mixing levels, split into three cases, namely, same gender female (SGF), same gender male (SGM), and different gender (DG). In general, the SNR between reference and estimated signal increases for the target and decreases for the masker as the mixing level in TMR increases. Further, the methods can be divided in those estimating the gain for both component signals simultaneously, and those estimating the gain independently for each component signal. The maximum likelihood based method, as multi-variable estimation method, is insensitive to TMR changes (see Figure 5.5 (c)). The nonlinear, the matching pursuit and the auditory gain estimation as representatives of the singular variable estimation procedure, show in general lower TMR improvements. For these methods, the gain estimation performance decreases for the masker with increasing TMR. This is expected, since the estimation problem gets more difficult the more speaker specific energy is obscured. As a comparison, without gain estimation, we measure a TMR of, e.g. 1.58 dB for the target and 2.55 dB for the masker speaker for the SGF case and an equal mixing level.

Maximum-likelihood based gain estimation results:



(a) Speaker FE1

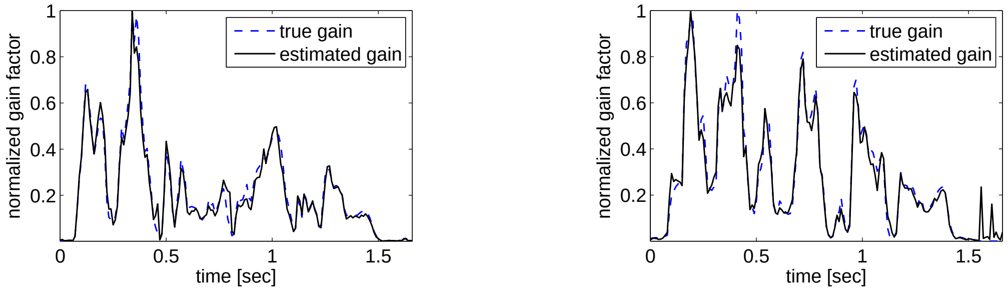
(b) Speaker FE2

(c) Gain estimation performance for four different mixing levels measured in TMR.

TMR	SGF		SGM		DG	
[dB]	t	m	t	m	t	m
0	29.02	24.79	27.18	25.17	23.44	25.17
3	29.38	27.79	26.12	24.81	24.96	23.83
6	29.70	26.34	26.66	23.53	26.13	23.18
9	30.39	25.48	27.14	23.62	27.20	21.99

Figure 5.5: Maximum-likelihood gain adjustment. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases with different mixing levels for target (t) and masker (m) speaker.

Nonlinear gain estimation results:



(a) Speaker FE1

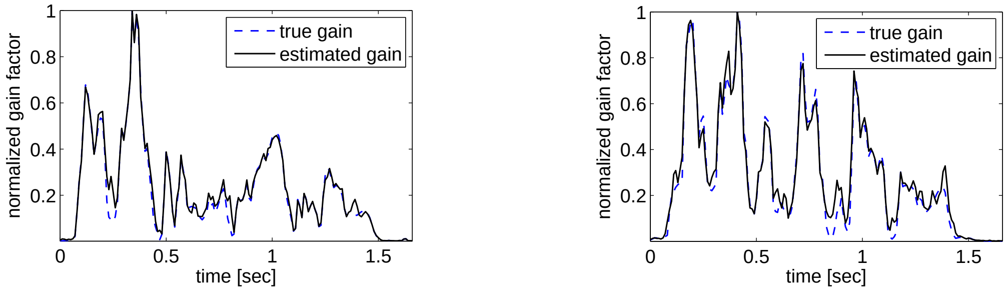
(b) Speaker FE2

(c) Gain estimation performance for four different mixing levels measured in TMR.

TMR	SGF		SGM		DG	
[dB]	t	m	t	m	t	m
0	17.1	19.1	15.7	16.1	19.3	18.2
3	17.5	19.1	15.9	12.9	19.6	16.1
6	18.6	16.5	16	11	19.4	13.7
9	20.8	13.6	16.3	8.6	19.3	10.5

Figure 5.6: Nonlinear gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker.

Projection-based gain estimation:



(a) Speaker FE1

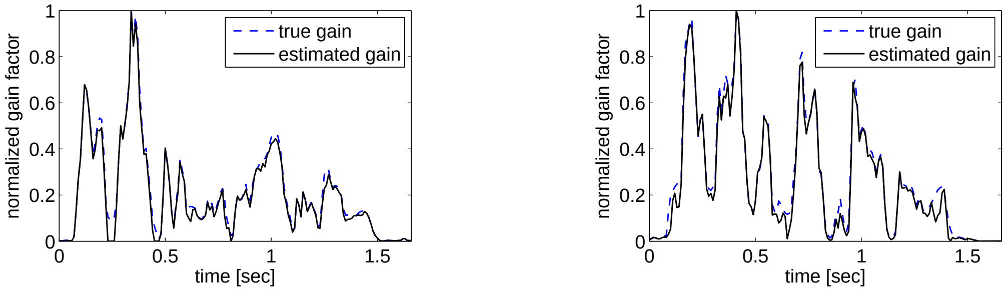
(b) Speaker FE2

(c) Gain estimation performance for four different mixing levels measured in TMR.

TMR	SGF		SGM		DG	
[dB]	t	m	t	m	t	m
0	22.05	15.90	10.60	12.90	18.70	14.44
3	25.91	12.95	14.33	9.84	22.34	9.76
6	29.43	10.43	17.51	7.14	25.80	6.18
9	31.94	8.13	28.88	4.20	20.99	4.89

Figure 5.7: Matching pursuit gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker.

Least Squares based gain estimation:



(a) Speaker FE1

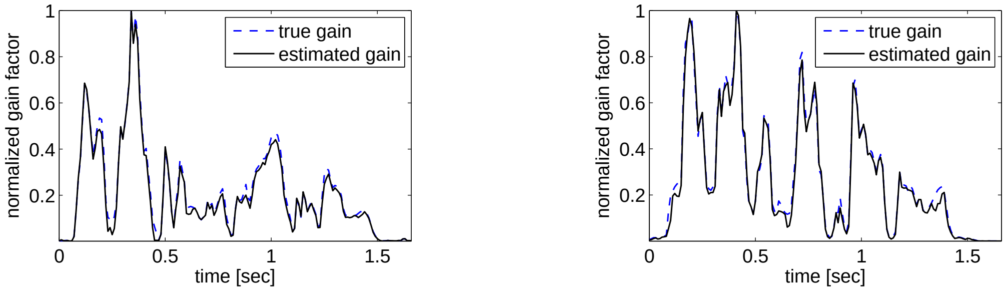
(b) Speaker FE2

(c) Gain estimation performance for four different mixing levels measured in TMR.

TMR	SGF		SGM		DG	
[dB]	t	m	t	m	t	m
0	24.07	20.82	13.46	19.09	21.76	24.42
3	25.58	18.69	16.05	16.74	23.99	22.35
6	27.07	17.01	18.86	14.35	26.36	20.20
9	28.70	15.60	21.49	11.97	28.68	18.14

Figure 5.8: Nonnegative least squares gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker.

Nonnegative Matrix Factorization based gain estimation:



(a) Speaker FE1

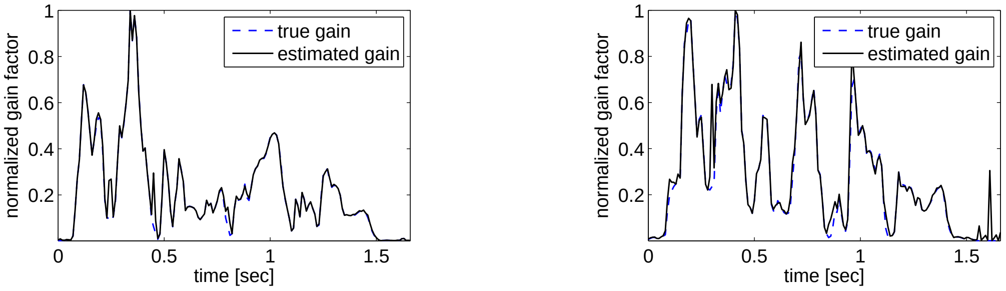
(b) Speaker FE2

(c) Gain estimation performance for four different mixing levels measured in TMR.

TMR	SGF		SGM		DG	
[dB]	t	m	t	m	t	m
0	25.00	23.44	17.86	18.32	21.42	22.78
3	26.54	21.53	19.08	18.97	22.97	21.61
6	27.52	20.34	20.77	16.59	20.77	16.59
9	28.66	19.02	22.13	15.29	26.60	19.01

Figure 5.9: Nonnegative matrix factorization based gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker.

Auditory motivated gain estimation:



(a) Speaker FE1

(b) Speaker FE2

(c) Gain estimation performance for four different mixing levels measured in TMR.

TMR	SGF		SGM		DG	
[dB]	t	m	t	m	t	m
0	24.73	20.22	15.88	20.68	21.85	13.85
3	28.37	16.97	20.26	16.75	25.01	9.92
6	31.75	13.77	23.78	13.05	27.74	7.58
9	33.71	10.73	26.39	9.72	29.98	5.40

Figure 5.10: Auditory motivated gain adjustment method. (a) and (b) show the true (blue dashed line) and estimated (black solid line) normalized gain for the component signals FE1 and FE2 observing just their mixture signal. (c) illustrates results for the three mixing cases at different mixing levels for target (t) and masker (m) speaker.

5.4.2 Shape-Gain Single Channel Source Separation

This section evaluates the performance of the introduced gain estimation methods for the SCSS task. In this experiment, all signals are mixed at equal level, using the experimental setup introduced in chapter 1, section 1.5. Moreover, the training speech material in the magnitude frequency domain has been normalized to unit norm prior to model training. The factorial-max VQ model with gain estimation is employed for source separation.

We investigate all three structures for the maximum-likelihood based gain estimation method: (i) The gain-shape decoder of Figure 5.3 denoted as *GS-VQ* method. This decoder performs gain estimation of each shape combination firstly. Afterwards, the shape models are adjusted by the gain estimates. Secondly, these gain adjusted models are used to find the most likely shape combination. (ii) The iterative shape-gain structure of Figure 5.2 is labeled as *SG-VQ*. As a first step, this decoder determines the shapes by the projection measure. Afterwards, the gains are just estimated for selected shapes. (iii) The ML-based gain estimation method is adapted for singular value optimization. Hence, the independent gain estimation decoder, introduced in section 5.3.3, is used for separation. This method is denoted as *Ind-ML-GE* in the experiment. Note, the gain estimation is carried out for every time step. Figure 5.11 compares the performance in TMR of the three structures. For all cases, the *Ind-ML-GE* decoder shows superior performance. For the remaining two simultaneous gain estimation methods, the shape gain VQ (*SG-VQ*) increases the TMR better than the gain-shape structure (*GS-VQ*). It is worth to note that for the ML-based gain estimation the independence of the hidden variables, associated to each speaker can also be employed for gain estimation. A comparison of the structures in terms of computational complexity shows a much lower complexity for the *SG-VQ*. While for the *SG-VQ* the gains are just estimated for selected states, the *GS-VQ* structure estimates the gains for all state combinations of the two speakers. This structure results in a Q-times higher complexity compared to the *SG-VQ* structure, i.e., $\mathcal{O}(\text{“GS-VQ”}) = \mathcal{O}(Q\ \text{“SG-VQ”})$, for the used ICM4 heuristic. Finally, the *Ind-ML-GE* gain estimation based separation method, using ICM4, increases the computational complexity compared to the ICM4 just marginally. Thus, this method combines both, good performance and low additional complexity.

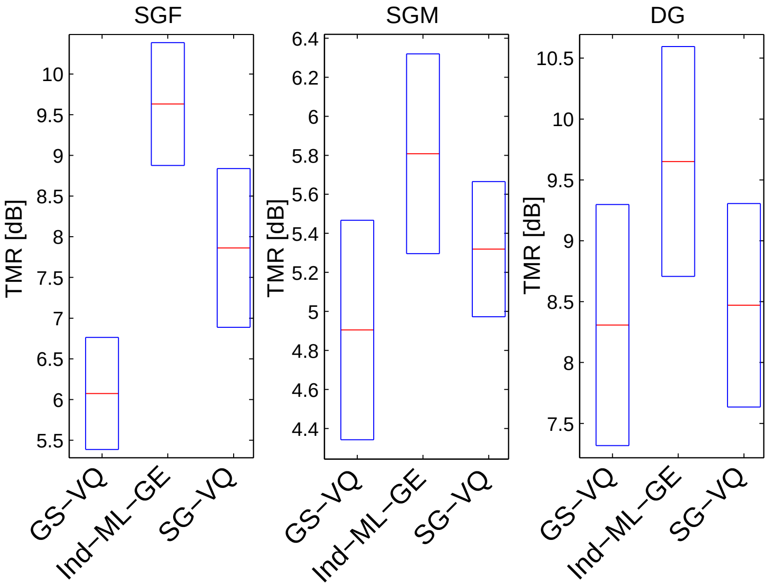


Figure 5.11: Separation performance in TMR for the ML-based gain estimation. Results are shown for gain-shape based (GS-VQ), independent gain estimation (Ind-ML-GE), and shape-gain (SG-VQ) based separation of two speech signals.

The next experiment investigates the nonlinear percentile filtering technique for gain estimation. We assess source separation performance of the percentile filtering method by taking different values for the r^{th}-percentile. Specifically, r ranges from the 0.1^{st}- to the 0.5^{th}-percentile in steps of 0.1. The gains of each speaker are estimated independently using the structure shown in Figure 5.4. Figure 5.12 depicts the results. The percentile technique using the 30 percent percentile improves the TMR the most for all 3 cases. Therefore, it is used for comparison to the other gain estimation methods.

In section 5.2.5, NMF was introduced for pure gain estimation. This experiment compares the independent gain decoder (*Ind-GE-NMF*), as singular value estimation and the gain-shape decoder (*GS-NMF-VQ*), as multi-value estimation method. Figure 5.13 depicts the performance, measured in TMR. Methods are identified by the labels. For the SGF case, the independent estimation outperforms *GS-NMF-VQ*, by more than 1 dB. In the DG case, the two methods show similar results. For the SGM case only, *GS-NMF-VQ* performs slightly better.

Figure 5.14 compares the separation performance in TMR of all discussed gain estimation methods. For all methods, the same speaker dependent shape VQ models are employed. The label *Ind-ML-GE* refers to the maximum likelihood based gain estimation using independent gain estimation (see Figure 5.4). *Perc-30* is the label for the percentile based gain estimation

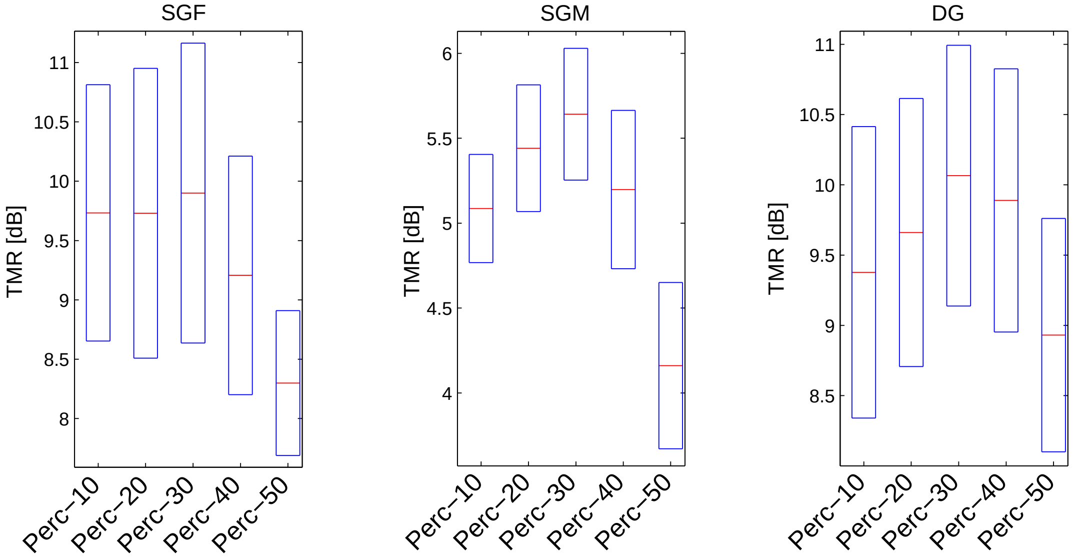


Figure 5.12: Separation results in TMR using nonlinear gain estimation. Performance is plotted for different percentiles, i.e. the 10, 20, 30, 40 and 50 percent percentile.

using the 30 percent percentile, *Projection* refers to the matching pursuit method, discussed in section 5.2.3 and *NNLS* to the least squares solution using nonnegative constraints, as introduced in section 5.2.4. Moreover, *Ind-GE-NMF* is used as NMF based method for gain estimation and *Mask-GE* identifies the auditory motivated gain estimation method. First of all, we notice that only singular value estimation techniques are amongst the methods. Moreover, for the whole separation process all except the NMF method can increase the TMR by approximately the same amount. The ML-based method *Ind-ML-GE* showed superior performance for the supervised gain estimation, discussed in section 5.4.1. In the separation experiment, however, separation performace does not exceed that of the other methods but is about the same.

Finally, we compare performance of the gain-shape separation methods to the separation performance of FM-VQ without gain estimation. We refer to the gain-shape source separation methods as gain-shape factorial-max VQ (*GS-FM-VQ*). Due to the additional gain estimation for the *GS-FM-VQ* methods we expect a better performance for the FM-VQ method for equal level speech mixtures. Performance comparison is made to the binary mask (*BM*) results of Figure 3.9 shown in chapter 3. Indeed, for the SGF case FM-VQ outperforms GS-FM-VQ by almost 2 dB TMR. However, for the DG case, the results are similar and for the SGM case, GS-FM-VQ shows superior performance. For this comparison we selected the *Ind-ML-GE* method as *GS-FM-VQ*.

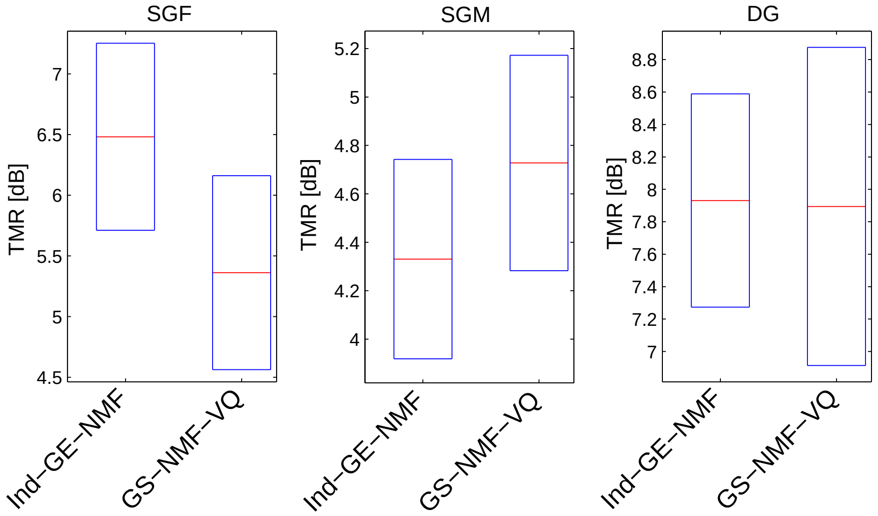


Figure 5.13: Separation results in TMR using nonnegative matrix factorization based gain estimation. Two structures are investigated for the shape estimation: (i) "Ind-GE-NMF": The gains are estimated independently. (ii) "GS-NMF-VQ": The gain-shape structure is employed for SCSS.

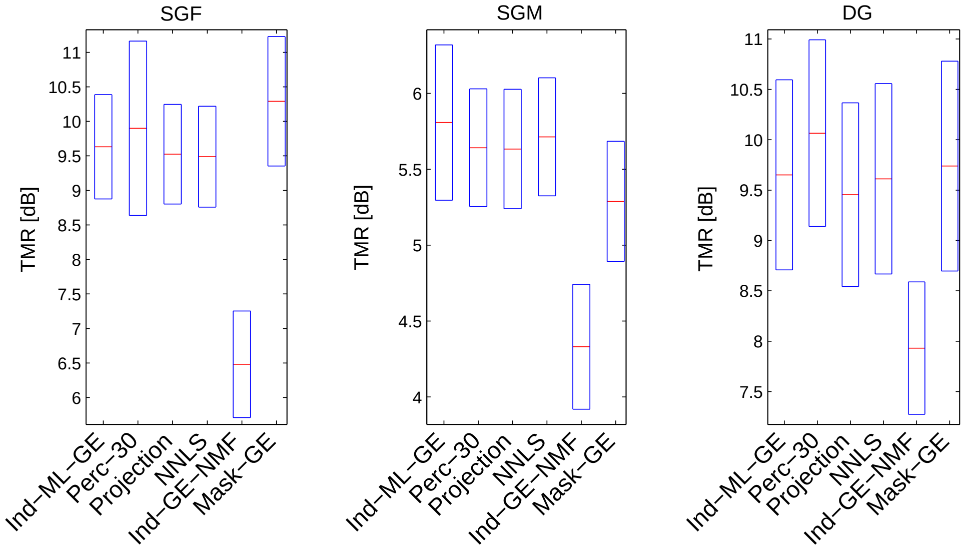


Figure 5.14: Separation performance comparison in TMR of introduced gain estimation methods. Performance is plotted for SGF, SGM and DG case separately.

5.5 Conclusion

This chapter addressed the problem of different mixing levels in single-channel source separation. For separation, speaker dependent statistical models are typically trained. These models perform best at equal level. Current approaches select a global TMR for an utterance from a discrete set, which is restrictive for online applications. Therefore, we proposed to estimate the gain of each component signal on a frame based level, as nonnegative matrix factorization does. This results in a gain-shape representation of the signal, as done in coding [50]. In order to find the best gain estimation method, we followed different strategies: We derived a maximum likelihood based solution for multi- and singular-variable estimation and developed nonlinear techniques, one based on percentile filtering and the other was motivated by the human auditory system. Moreover, matching pursuit, non-negative least squares and NMF strategies have been pursued for gain estimation. We evaluated the methods in two experiments. In the first, only gain estimation was carried out, while in the second, single channel source separation (SCSS) was performed using a gain-shape representation of the speaker dependent (SD) models. Three different decoder structures have been proposed for shape estimation in the SCSS experiments. For multi-variable estimation an iterative gain-shape decoder and a simultaneous gain-shape estimation decoder have been introduced. We employed an independent gain estimation decoder for singular-variable estimation. In the gain estimation experiment, the ML-based method outperformed all other methods and showed an almost constant performance for different mixing levels. In the gain-shape separation experiment, the singular-variable estimation methods using independent gain estimation showed superior performance compared to multi-variable estimation methods and decoder structures. Only the NMF based method shows inferior performance in terms of TMR. Finally, the introduced gain-shape estimation for source separation method showed only lower performance for the SGF case compared to the factorial-max VQ method without gain estimation. Since the mixing level is usually not known in advance, the application of gain estimation is preferable in any case.

Chapter 6

Source Specific Characteristic for SCSS

6.1 Introduction

In the previous sections, we mainly focused on explicit models, where individual source characteristics are stored, during the training phase. In this section, we combine explicit and implicit models. Implicit models try to mimic the ability of the human auditory system. Here, the mixture is a scene to be organized and particular extracted components are merged to form output streams of individual sources. Therefore, features like common on- and offsets, harmonicity and amplitude- and frequency modulations are extracted and considered for the signal separation [3].

Motivated by the decomposition of the speech mixture into distinct parts, as implicit models does [3], we propose to perform separation based on the factorization of the underlying speech signal, in a fine- and a coarse-spectral structure. This also matches the speech production model [121]. For speech, the excitation signal, produced by the vocal folds, mainly represents the fine spectral structure, whereas, the coarse spectral structure can be linked to the shaping of the vocal tract. This decomposition has been already employed by Gomez et al. [35] and Radfar et al. [76].

In [76], the fundamental frequency (f_0) or its perceived counter part, the pitch information [1] of each speaker, using a multi-pitch tracking method [26], represented the source-driven part. Afterwards, the estimated pitch of each speaker is used to synthesize an artificial excitation signal, representing the fine spectral structure. Source separation is enabled by the combination of the artificial excitation signal with the statistical representation of the vocal-tract filters (VTFs). The reader is referred to chapter 1, section 1.2.1 for a detailed discussion on this method. We extend this system and replace modules in section 6.2.1.

Gomez et al. [35] used a two-layer source-filter transformation model. This is employed for their spectral deformation model, where prediction is made from an elapsed to a present local time-frequency cell. Therefore, neighboring and past observations are employed to infer missing, i.e. masked, data. Missing data prediction in the context of single channel source separation (SCSS) has been also discussed in [101]. The thesis of Gomez has been already discussed in section 1.2.2 of chapter 1.

This chapter compares sequential and parallel source-filter based SCSS approaches. Basic principles are discussed in section 6.1.1. In section 6.2, two sequential methods are presented.

[1] We use the terms pitch and fundamental frequency interchangeably in this work. Thus, in this context pitch always refers to the fundamental frequency. This nomenclature better correlates with previous literature [27].

The first extracts pitch information in the source-driven part and the second utilizes statistical models to represent the source-driven part. Pitch estimation for source separation has already been used in [122, 123, 75]. For voiced speech, these methods can separate the signals quite well. This is achieved by assigning the energy of the pitch frequency component and its multiples, the harmonics, to the respective speaker. During unvoiced speech, however, pitch information is not available and separation is impossible. Moreover, these methods often suffer from speaker permutations, where single pitch values or whole pitch tracks are assigned erroneously to the wrong speaker. Permutations mainly occur during unvoiced speech sections, and due to close or crossing pitch tracks. In [29] the permutation problem was addressed by introducing speaker dependent models. To this end, they represented all combinations of pitch pairs of two speakers by statistical models. During decoding, two independent pitch tracks have been determined by tracking through this factorial model. An alternative approach proposed by Shao et al. [14] performs clustering as post processing to form speaker unique segments. The application of this approach maintains a speaker independent model. Thus, the speaker identity must not be known a priori.

In the subsequent section 6.2.2 we replace the multi-pitch tracking unit by an unit, which directly models the excitation signal. Note, in this model the cascade of two factorial models remains. This method goes beyond the source-filter decomposition of speech signals and separates the coarse- from the fine-spectral structure of the speech mixture. In the spectral whitened speech mixture only the fine structure of the individual speakers remains. Based on this decomposition, the fine spectral structure of the underlying signals can be estimated for given excitation models and the observed speech mixture. Unfortunately, there is no direct relation between the speech mixture and the single excitation signals. Therefore, an approximation is introduced and evaluated. As a last step, we investigate the integration of the gain-shape modeling of chapter 5 for source-filter models.

Finally, section 6.3 introduces the parallel separation of the fine- and coarse-spectral structure. Therefore, for each excitation and VTF model a hidden variable is defined. Thus, a factorial model with four hidden variables describes the mixture observation for every time instance. Note, the exponential time complexity of this model makes exact learning and inference intractable. Therefore, we propose a complete factorization, which is based on the sampling method discussed in chapter 4, section 4.5. Note, this parallel structure avoids the approximation needed in the sequential structure using excitation models.

6.1.1 Source- and Model-Driven Approach

In the source-filter model, the speech signal is described as an excitation signal that is shaped by the vocal tract, acting as a filter process. Hence, a speech segment $s_i(t)$ is the convolution of the excitation $e_i(t)$, with the VTF response $h_i(t)$, which is further multiplied by a gain factor $a_i(t)$ in the time domain as:

$$s_i(t) = a_i(t)\ \big(e_i(t) \star h_i(t)\big), \tag{6.1}$$

where the speaker index is given as $i \in \{1, 2\}$. The convolution results in a multiplicative relation in the frequency domain and an additive relation in the logarithmic frequency domain as:

$$\mathbf{S}_i = \log a_i + (\mathbf{E}_i + \mathbf{H}_i). \tag{6.2}$$

This speech production model is depicted in Figure 6.1. The switch **sw** changes the excitation between voiced and unvoiced. For voiced excitation, an impulse train with period T_0 is generated. The period T_0 and the fundamental frequency f_0 are related as $T_0 = \frac{1}{f_0}$. The

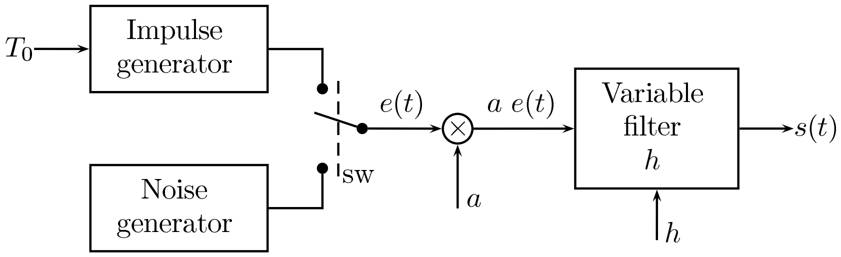


Figure 6.1: Speech production Model [adapted from: [121]].

gain a controls the power of the excitation signal. For source-filter based SCSS the gain is usually absorbed either in the excitation or the filter response h, thus, is modeled implicitly. We follow this strategy in a first step, but consequently model the gain independently, in a second separate step. For this task, the methods introduced in chapter 5 will be utilized.

For the proposed models which directly model the excitation signals, the source-filter representation of Eq. (6.2) can be used to adapt the *mixmax* interaction model of chapter 3, Eq. (3.16). Therefore, a hidden random variable v_i is introduced for the VTF related model and a random variable u_i for the excitation related model. We assume that both variables are a set of U, $v_i, u_i \in U$, where U has cardinality $Q = |U|$. In this model, the emission densities $\phi_s(z)$ of the model are replaced by the emission densities of the fine- and coarse-spectrum related random variables. Following the above discussion, the observation model can be adapted to:

$$p(\mathbf{Y}|z_1, z_2) = \mathcal{N}(\mathbf{Y}; \max(\mu_s(z_1), \mu_s(z_2)), \Sigma), \tag{6.3}$$

where

$$\mu_s(z_i) = \log a_i + \mu_e(u_i) + \mu_h(v_i) \tag{6.4}$$

is the superposition of the means of the excitation ϕ_e and filter ϕ_h emission densities. The variance can be found in a similar way.

6.2 Sequential Source-Filter based SCSS

In the sequential source-filter based SCSS model, two factorial models are cascaded. In the first model, the excitation related signals are estimated whereas in the second, the estimated excitations are employed for VTF estimation. For excitation estimation, two strategies are pursued: In section 6.2.1, a pitch track associated to each speaker is extracted using a probabilistic double-pitch tracking method [29]. Following, an artificial excitation signal is synthesized based on the pitch estimate.

In section 6.2.2, both the excitation signal and vocal tract filter are represented by a statistical model. For signal separation, two factorial models are concatenated. The first factorial model is employed to represent the spectral whitened speech mixture as observation. Afterwards, the means of the emission densities associated with the determined state sequences are used to estimate the corresponding VTF state sequences. Finally, the state emission densities of the most likely state sequence of each speaker are combined using Eq. (6.4). This final sequence of state emission means is employed for mask estimation.

6.2.1 Synthetic Excitation Modeling[1]

The overall multi-pitch source-filter SCSS system is shown in Figure 6.2 and consists of the following building blocks: A multi-pitch tracking unit followed by the excitation generation unit is representing the source-driven part. We compare speaker dependent (SD), gender dependent (GD) and speaker independent (SI) multi-pitch tracking performance and employ them for speech separation. Once the pitch trajectories of each speaker are estimated, i.e. f_0^1 and f_0^2, they are further utilized to create the excitation signals $\mathbf{E}_1$ and $\mathbf{E}_2$. VTFs, known as spectral envelopes, are extracted in a speaker independent way from training data s^{train} and are used to train both, VQ and NMF models, i.e. $\lambda_{\text{SI}}^{\text{VQ}}$ and $\lambda_{\text{SI}}^{\text{NMF}}$. Note, in this section we skip the superscript H to identify a VTF model λ^H, instead we clearly mask the two considered VTF models, i.e. λ^{NMF} and λ^{VQ}. The combination of the excitation signal $\mathbf{E}_i$ and the VTF model, which is carried out in the model construction block of Figure 6.2, results in an utterance dependent (UD) model $\lambda_{\text{UD}}^{\text{VQ}}$ or $\lambda_{\text{UD}}^{\text{NMF}}$, i.e. the VTFs in combination with the excitation are modeling a particular utterance. Thus, the harmonic excitation signal acts as discriminative feature and introduces utterance dependency, which enables speech separation. The UD model is further used for speech separation, performed in the separation step.

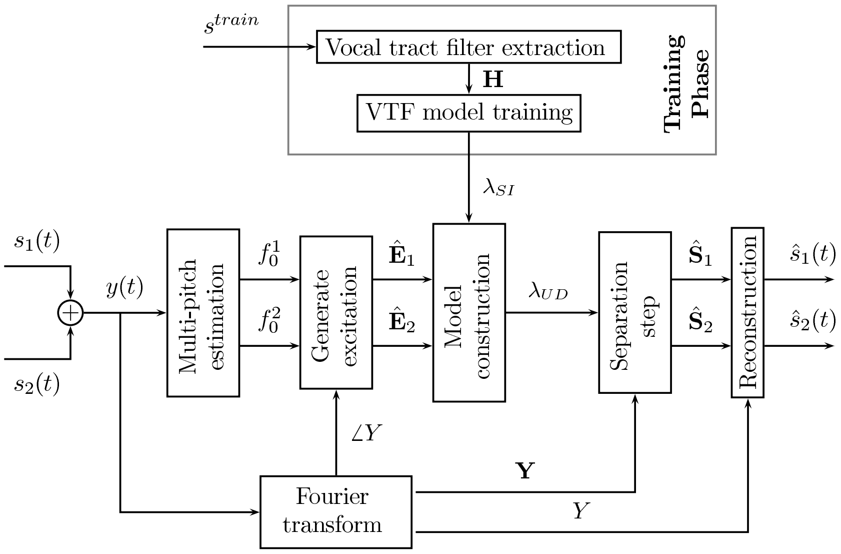


Figure 6.2: Blockdiagram of the separation system.

For performance analysis the component signals are recovered in two ways:

- The most likely state sequences of the UD model of each component speech signal are used to find the respective binary masks (BMs). Afterwards the BM is used to filter the speech mixture in order to get an estimate of the component signal $\hat{\mathbf{S}}_i$.
- The emission means of the most likely state sequence of the UD model are directly used for synthesis of the component speech signals $\hat{\mathbf{S}}_i$.

In the reconstruction block of Figure 6.2, the separated speech signals are synthesized by first applying the inverse Fourier transform on each speech segment using the phase of the mixed speech signal $\angle Y$. For speech signal reconstruction the overlap-add method is used.

[1] M. Stark, M. Wohlmayr and F. Pernkopf. Source-Filter Based Single Channel Speech Separation Using Pitch Information. In *IEEE Transactions on Audio, Speech and Language Processing (TASLP)*, accepted for publication.

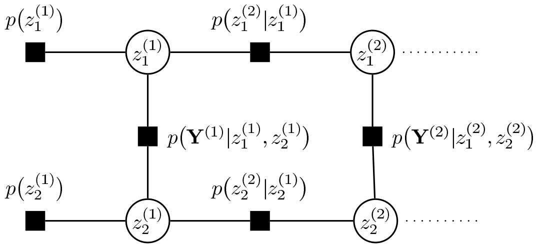


Figure 6.3: A factorial HMM shown as a factor graph [1]. Factor nodes are shown as shaded rectangles together with their functional description. Hidden variable nodes are shown as circles. Observed variables $\mathbf{Y}^{(\tau)}$ are absorbed into factor nodes.

In this work, we use different models to represent certain speaker spaces. The SI space is characterized by one universal model, valid for all speakers and phonemes they can articulate. The GD model is trained to represent the distribution unique for each gender, male or female. Further, the SD model describes the space of each individual speaker. A subset of the SD space is the utterance dependent space, i.e. an individual model per utterance. Hence, the SI space can be decomposed according to: UD $\subseteq$ SD $\subseteq$ GD $\subseteq$ SI.

Multi-pitch Tracking using F-HMM[1]

We use a F-HMM for tracking the pitch trajectories of both speakers. The F-HMM represented as factor graph [1] is shown in Fig. 6.3. The hidden state random variables are denoted by $z_i^{(\tau)}$, where $i \in \{1, 2\}$ indicates the Markov chain related to the speaker index and τ, the time index from 1 to T. Similarly, the observed random variables, i.e. the log magnitude spectrum, are denoted by $\mathbf{Y}^{(\tau)}$ at time τ. Each $z_i^{(\tau)}$ represents a discrete random variable related to the pitch of speaker i at τ, while $\mathbf{Y}^{(\tau)}$ is continuous. For simplicity, all hidden variables are assumed to have cardinality $|Z|$. The edges between nodes indicate a conditional dependency between random variables. Specifically, the dependency of hidden variables between two consecutive time instances is defined for each Markov chain by the transition probability $p(z_i^{(\tau)}|z_i^{(\tau-1)})$. The dependency of the observed variables $\mathbf{Y}^{(\tau)}$ on the hidden variables of the same time frame is defined by the observation probability $p(\mathbf{Y}^{(\tau)}|z_1^{(\tau)}, z_2^{(\tau)})$. Finally, the prior distribution of the hidden variables in every chain is denoted by $p(z_i^1)$. Denoting the whole sequence of variables, i.e. $\{z^{(\tau)}\} = \bigcup_{t=1}^{T}\{z_1^{(\tau)}, z_2^{(\tau)}\}$ and $\{\mathbf{Y}^{(\tau)}\} = \bigcup_{t=1}^{T}\{\mathbf{Y}^{(\tau)}\}$, the joint distribution of all variables is given by

$$\begin{gathered}p(\{z^{(\tau)}\}, \{\mathbf{Y}^{(\tau)}\}) = p(\{\mathbf{Y}^{(\tau)}\}|\{z^{(\tau)}\})p(\{z^{(\tau)}\}) = \\ \prod_{i=1}^{2}\Big[\ p(z_i^1)\prod_{\tau=2}^{T} p(z_i^{(\tau)}|z_i^{(\tau-1)})\ \Big]\prod_{\tau=1}^{T} p(\mathbf{Y}^{(\tau)}|z_1^{(\tau)}, z_2^{(\tau)}).\end{gathered}$$

The number of possible hidden states per time frame is $|Z|^2$. As pointed out in [93], this could also be accomplished by an ordinary HMM. The main difference, however, is the constraint placed upon the transition structure. While an HMM with $|Z|^2$ states would allow any $|Z|^2 \times |Z|^2$ transition matrix between two hidden states, the F-HMM is restricted to two $|Z| \times |Z|$ transition matrices.

[1] This section is included with kind permission of Michael Wohlmayr and Franz Pernkopf.

F-HMM Parameters: The state-conditional observation likelihoods $p(\mathbf{Y}^{(\tau)}|z_1^{(\tau)}, z_2^{(\tau)})$ are modeled with a GMM, using $M \geq 1$ components according to

$$p\left(\mathbf{Y}^{(\tau)}|\mathbf{\Theta}_{z_1,z_2}\right) = \sum_{m=1}^{M} \alpha_{z_1,z_2}^m \, \mathcal{N}\left(\mathbf{Y}^{(\tau)}; \mathbf{\Theta}_{z_1,z_2}^m\right).$$

To obtain $\mathbf{Y}^{(\tau)} \in \mathbb{R}^{64}$, we first apply the zero padded 1024 point FFT on a Hamming windowed signal segment $y(t)$ of length 32ms. Next, we take the log magnitude of spectral bins 2-65, which corresponds to a frequency range up to 1 kHz. This covers the most relevant frequency range of resolved harmonics while keeping the model complexity low. α_{z_1,z_2}^m corresponds to the weight of each component $m = 1, \ldots, M$. These weights are constrained to be positive $\alpha_{z_1,z_2}^m \geq 0$ and $\sum_{m=1}^{M} \alpha^m = 1$. The parameters $\mathbf{\Theta}_{z_1,z_2} = \left\{\alpha_{z_1,z_2}^m, \mathbf{\Theta}_{z_1,z_2}^m\right\}_{m=1}^{M}$ can be learned by the EM algorithm [124], where $\mathbf{\Theta}_{z_1,z_2}^m = \left\{\boldsymbol{\mu}_{z_1,z_2}^m, \mathbf{\Sigma}_{z_1,z_2}^m\right\}$.

Each hidden variable has $|Z| = 200$ states, where state value '1' refers to 'no pitch', and state values '2'-'200' correspond to different pitch periods ranging from less than 1ms to 12.5ms, i.e. from $\sim$ 1 kHz down to 80 Hz. Note that segments of silence and unvoiced speech are modelled by $z = 1$. For learning the GMM, we need *supervised* data, i.e. the pitch-pairs for the corresponding speech mixture spectrograms. These data are composed from single speaker recordings, using the $RAPT$ pitch extraction [2]. Hence, with both pitch trajectories for the mixed utterances at hand, we can easily learn a GMM $p\left(\mathbf{Y}^{(\tau)}|\mathbf{\Theta}_{z_1,z_2}\right)$ for each pitch-pair (z_1, z_2). Accordingly, we have to determine 200 × 200 GMMs. Unfortunately, data might be rarely available for some pitch-pairs, whereas, there is plenty of data for, e.g. $(z_1 = 1, z_2 = 1)$. For this reason, we use the minimum description length (MDL) criterion [125] to determine the number of components of the GMM automatically. The MDL criterion [126] is

$$MDL = -\log p\left(\mathbf{Y}_{z_1,z_2}|\mathbf{\Theta}_{z_1,z_2}\right) + \frac{M\,(L+1)}{2} \log N_{z_1,z_2},$$

where L is the number of parameters per component (for GMMs with diagonal covariance matrix $L = 2d$ where $d = 64$ in our case), in $\mathbf{Y}_{z_1,z_2}$ all spectrogram samples belonging to (z_1, z_2) are collected, and N_{z_1,z_2} denotes the size of $\mathbf{Y}_{z_1,z_2}$. This equation has the intuitive interpretation that the log-likelihood $-\log p\left(\mathbf{Y}_{z_1,z_2}|\mathbf{\Theta}_{z_1,z_2}\right)$ is the code length of the *encoded* data. The term $\frac{M(L+1)}{2} \log N_{z_1,z_2}$ models the optimal code length for all parameters $\mathbf{\Theta}_{z_1,z_2}$. In case of $N_{z_1,z_2} = 1$, for a particular (z_1, z_2) we use a single Gaussian with $\boldsymbol{\mu}_{z_1,z_2}^m = \mathbf{Y}_{z_1,z_2}$ and $\mathbf{\Sigma}_{z_1,z_2}^m$ is set to a small $\sigma_{min}\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. For $N_{z_1,z_2} > 1$, we train GMMs with M ranging from 1 to 15, and take the GMM whose corresponding MDL criterion is minimal. If there is no training sample available for the pitch-pair (z_1, z_2), i.e. $N_{z_1,z_2} = 0$, we set $\boldsymbol{\mu}_{z_1,z_2}^m = 0$ and $\mathbf{\Sigma}_{z_1,z_2}^m = \mathbf{I}$. Prior to pitch tracking all spectrogram samples are normalized to zero mean and unit variance. Finally, we multiply the pitch likelihood $p(\mathbf{Y}^{(\tau)}|z_1^{(\tau)}, z_2^{(\tau)})$ with the pitch-pair prior $p(z_1^{(\tau)}, z_2^{(\tau)})$, since this slightly improved the performance in our experiments.

Both transition matrices of the F-HMM $p(z_i^{(\tau)}|z_i^{(\tau-1)})$ are obtained by counting and normalizing the transitions of the pitch values from single speaker recordings. Additionally, we apply Laplace smoothing[2] on both transitions $p(z_i^{(\tau)}|z_i^{(\tau-1)})$. The prior distributions $p(z_i^1)$ are obtained in a similar manner.

[2] Laplace smoothing amounts to the initialization of each element of the transition matrix with count one, i.e. adding the prior information that each transition was observed at least once. This smoothes the transition probabilities.

Tracking: The task of tracking involves searching the sequence of hidden states $\{z^{(\tau)}\}^*$ that maximizes the conditional distribution $p(\{z^{(\tau)}\}|\{\mathbf{Y}^{(\tau)}\})$:

$$\{z^{(\tau)}\}^* \quad = \quad \underset{\{z^{(\tau)}\}}{\operatorname{argmax}}\, p(\{z^{(\tau)}\}|\{\mathbf{Y}^{(\tau)}\}) \tag{6.5}$$

For HMMs, the exact solution to this problem is found by the Viterbi algorithm. For F-HMMs, an exact solution can be found using the junction tree algorithm [94]. However, this approach gets intractable with increasing number of hidden Markov chains and $|Z|$. Algorithms for approximate and exact solutions on F-HMMs are derived in [93]. Approximate inference algorithms are often derived from the framework of variational inference. The sum-product algorithm [1] can be derived under a similar setting of variational principles [127], although more intuitive derivations exist for graphs without loops. When applied on a graph with loops, as is the case for a F-HMM, the solutions are in general not guaranteed to converge and can only approximate the optimal solution.

For multi-pitch tracking, we explored the max-sum algorithm (a variant of sum-product algorithm) as well as the junction tree algorithm. We apply both variants on the *loopy* F-HMM graph to obtain a solution for Eq. (6.5). In contrast to the junction tree algorithm, the max-sum algorithm can only approximate Eq. (6.5). In [29], experimental results suggested that the obtained solutions sufficiently approximate the exact solution, while computational complexity is much lower. Indeed, the time complexity of the max-sum algorithm applied to a F-HMM is $\mathcal{O}(TK|z|^K)$, where K is the number of Markov chains. In contrast, the time complexity of the junction tree algorithm is $\mathcal{O}(TK|z|^{K+1})$.

In the sequel, we give a short overview of the used max-sum message passing algorithm. For a detailed discussion, we refer the interested reader to [1, 94, 127]. Further, details on the junction tree algorithm are given in [93]. The max-sum algorithm is based on passing messages between nodes of a graph. Among various types of graphs, factor graphs [1] have become popular to depict the mechanisms of message passing. Figure 6.3 shows a F-HMM as factor graph, where the functional dependency of each variable node, for brevity called z, is made explicit by "factor nodes", shown as shaded rectangles, i.e. each rectangle denotes a function $f(\{\hat{z}\})$ of its adjacent (i.e. neighboring) variable nodes $\{\hat{z}\}$.

For the max-sum algorithm, each node sends to every neighbor a vector valued message μ, which is itself a function of the messages it received, (as well as $f(\{\hat{z}\})$, for the case of a factor node). A message from variable node z to factor node f is

$$\mu_{z\to f}(z) \quad = \quad \sum_{g\in n(z)\backslash f} \mu_{g\to z}(z), \tag{6.6}$$

while a message from factor f to variable z is

$$\mu_{f\to z}(z) \quad = \quad \max_{\{\hat{z}\}\backslash z} \left(\ln f(\{\hat{z}\}) + \sum_{y\in\{\hat{z}\}\backslash z} \mu_{y\to f}(y) \right). \tag{6.7}$$

Here, $n(z)$ denotes the set of neighbor nodes of z. We normalize each message and restrict each node to send a maximum of 15 messages per link. Further, each node only re-sends a message to a neighbor if it is significantly different from the previously sent message in terms of the Kullback-Leibler-divergence [106]. After the last iteration, we obtain the maximum *a posteriori* configuration $p^*(z)$ of each variable node $z \in \{z^{(\tau)}\}$ as a function of its incoming messages according to

$$p^*(z) = \max_{\{z^{(\tau)}\}\backslash z} p(\{z^{(\tau)}\}|\{\mathbf{Y}^{(\tau)}\}) = \sum_{g\in n(z)} \mu_{g\to z}(z). \tag{6.8}$$

Although the set of maxima $z^* = \text{argmax}_z\, p^*(z)\ \forall z \in \{z^{(\tau)}\}$ does not necessarily yield the global maximum $\{z^{(\tau)}\}^*$, as multiple local maxima might be present, a backtracking stage may lead to inconsistencies due to the loops in the factor graph. For this reason, we simply set the global maximum $\{z^{(\tau)}\}^*$ to the set of individual maxima z^*.

Excitation Synthesis: Once the pitch tracks are estimated for each speaker, the harmonic part of the excitation signal is modeled as:

$$e_i(t_f, \omega_0^i, \angle\, Y(u)\ ) = \sum_{u=1}^{U(\omega_0^i, f_{max})} \sin(\ u\ \omega_0^i t_f\ +\ \angle\, Y(u)\ ), \tag{6.9}$$

where U denotes the number of harmonics up to a specified highest frequency f_{max} set to 4kHz, ω_0 is f_0 in radians, $\angle\, Y$ is the phase of the mixed signal and $t_f = [1, \ldots, T_f]$ is the time index of a frame. ω_0^i at t is the sampling frequency divided by $z_i^{(\tau)}$, i.e. $\omega_0^i = 2 \cdot \pi \frac{fs}{z_i^{(\tau)}}$ for all $2 \leq z_i^{(\tau)} \leq 200$ states. For unvoiced signals, i.e. $z_i^{(\tau)} = 1$, a Gaussian random signal is used as excitation. To the voiced signals, a Gaussian random signal filtered by a high-pass with cut-off frequency at f_{max} is added to Eq. (6.9). This equation is similar to the harmonic plus noise model [128] or sinusoidal modeling [129] but without amplitude weighting of the harmonics. In the source-filter system, this weighting is provided by the VTF estimation algorithm.

Vocal Tract Filter Models

For the VTF estimation the excitation signals $\mathbf{E}_1$ and $\mathbf{E}_2$ are assumed to be *a priori* known from multi-pitch tracking. The synthesized excitation signals are employed to construct utterance dependent models as defined in Eq.(6.3). In this section, we investigate two different statistical VTF models for speech separation. The first method is based on the maximum likelihood (ML) estimation of the VQ codewords. During decoding the mixture maximization (*mixmax*) approach [95] is used as interaction model to represent the speech mixture $\mathbf{Y}$. Moreover, we incorporate the nonlinear gain estimation method discussed in chapter 5, section 5.2.2 to make the separation approach suitable for different mixing levels. Finally, we restrict the search space of the VQ to the most promising codewords by applying the techniques of restricted observation likelihood calculation of chapter 4, section 4.4. Second, we examine non-negative matrix factorization to model the VTFs. In contrast to the VQ VTF model, NMF provides the advantage that the gain of each basis is inherently determined by the update rules.

To extract the vocal tract filters, we use the spectral envelope vocoder (SEEVOC) method described in [130, 131]. The SEEVOC method operates in the magnitude or log-magnitude frequency domain to extract the spectral envelope. Basically, the method determines and marks local maxima. For maxima determination, SEEVOC operates in two modes. In the voiced mode, the predetermined pitch value is utilized to detect local maxima in the vicinity of the fundamental frequency and its multiples, the harmonics. In the unvoiced mode, local maxima are determined using a minimum frequency range criterion. Peaks in this range are excluded as peak candidates. Once all local maxima are determined, the envelope of the VTF is calculated using a cubic spline interpolation. The excitation signal is calculated by division or subtraction of the VTF from the speech signal, either in the magnitude or log-magnitude domain, respectively. Our SEEVOC implementation is operating in the log-magnitude domain and uses the *RAPT* pitch extraction [2] algorithm.

The careful reader may have recognized that the gain information is implicitly included in the VTFs $\mathbf{H}$. Consequently, the excitation signal is normalized such that the maximum over frequency has a value of 0 dB in the log-magnitude domain. For voiced signals, the excitation has a comb-like structure with flat spectrum. Unvoiced excitation has a noise like spectrum. Exactly this normalization property of the excitation signal in the SEEVOC method enables the replacement of the true excitation signal with the artificial one, introduced in section 6.2.1.

In this method, for an equal mixing level of two speech signals the gain factor can be excluded from the model in Eq. (6.2) or set to $a_{z_i} = 1$. For different mixing levels, however, the model does not match anymore and has to be adjusted. Since VQ is prone to model the same VTF shapes at different gain levels with separate bases $\mu^h_{v_i}$, training data are mean normalized for the system with gain estimation. This normalization reduces model complexity and increases robustness in model learning.

Separation Using Non-Negative Matrix Factorization: Alternatively, we have investigated NMF [65, 71] for VTF modeling. NMF approximates a non-negative matrix $V^{D\times T}$ by the product of two non-negative matrices $W^{D\times R}$ and $A^{R\times T}$, where D is the number of frequency bins and R is the approximation level, i.e. the number of bases. In our case, the VTF training data in the magnitude frequency domain corresponds to $V = \mathbf{h}_{\text{train}}$. The SI bases W are estimated and collected in $\lambda^{\text{NMF}}_{\text{SI}}$. The decomposition of V, in W and A is based on minimizing the Kullback-Leibler distance [65]. The bases W are estimated during the training, whereas the weights are estimated in the separation phase. These weights specify the contribution of each basis for the approximation of the speech mixture $\mathbf{y}$. Typically, in the separation step, a union of all UD bases is carried out by combining them as $W_{\text{UD}} = W_{\text{UD}_1} \cup W_{\text{UD}_2}$. The UD bases W_{UD_i} can be constructed from W using the excitation $\mathbf{e}_i$ as:

$$W_{\text{UD}_i} = W \cdot \mathbf{e}_i.$$

During separation, we fix the bases W_{UD} and estimate the weights A to achieve the best approximation of the mixed signal $\mathbf{y}$. Further, the reconstruction is done by first splitting the bases matrix W_{UD} as well as the estimated weight matrix A into the parts belonging to the corresponding sources. Finally, the reconstruction of the signals is given as:

$$\hat{\mathbf{s}}_i = W \ \mathbf{e}_i \ A_i,$$

where $\hat{\mathbf{s}}_i$ is the respective estimated spectrum of speaker i.

6.2.2 Direct Statistical Excitation Modeling[1]

In the previous section a multi-pitch estimation method was employed to estimate the pitch, which was used to synthesize an excitation related signal. This section replaces the multi-pitch method and directly represents the excitation signals by a factorial model. In the previous section 6.2.1 single speaker utterances have been divided into their fine- and coarse-spectral structure for model training only. For decoding, the speech mixture has been investigated for both, multi-pitch tracking and VTF estimation. In this section, the sequential separation structure remains, but the excitation signals are estimated directly. As a consequence, the

[1] This section is an updated and extended version of the paper:
M. Stark and F. Pernkopf. Towards source-filter based single sensor speech separation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 97-100, Taipei, Taiwan, April 2009

speech mixture can not be directly used as observation. Instead, a spectrally whitened speech mixture is expected as observation for the new model. Therefore, also the speech mixture has to be split into its fine- and coarse-spectral structure. In contrast to single speech utterances, the spectrally whitened speech mixture has no physical meaning. Moreover, there is no direct relation between the spectrally whitened speech mixture and the component excitation signals:

$$y = e_y(t) \star h_y(t) = (e_1(t) \star h_1(t)) + (e_2(t) \star h_2(t)), \tag{6.10}$$

where $e_y(t)$ is the spectrally whitened speech mixture and $h_y(t)$ its spectral envelope. However, we will show that the spectrally whitened speech mixture approximates the sum of the individual excitation signals sufficiently well.

To include prior knowledge about the respective excitation signals, both, a VQ and an HMM are employed. An HMM also captures time dependencies and, hence, should reduce permutations among both speakers. The speaker dependent trained models are combined in an FM-HMM or FM-VQ in order to estimate the excitation signals from the speech mixture. Having an estimate related to the excitation signal at hand, the coarse spectral structure can be estimated similarly to an analysis-by-synthesis procedure in speech coding [121], where a trained VQ codebook, models the vocal-tract prior knowledge. For vocal-tract modeling we investigate models, with and without dynamics.

The structure for model training is shown in Figure 6.4. Before training, the speech signal is divided into source and filter related parts using the linear prediction technique (LPC) [132, 121]. For every speech segment $\tau = [1, \dots, T]$ source and filter are related as

$$s_i(t) = -\sum_{n=1}^{N} c_n \cdot s_i(t-n) + e_i(t),$$

where N specifies the filter order, c_n the filter coefficients, and $e_i(t)$ denotes the residual or excitation signal in the time domain. The excitation signal and the vocal tract filter of each speaker are transformed to the log-frequency domain $\mathbf{E}_i$ and $\mathbf{H}_i$. Afterwards, the training data is used to find a statistical representation, either an HMM λ_i^{HMM} or an VQ λ_i^{VQ}. In order to enhance filter stability and increase robustness against quantization errors, the LPC coefficients can be represented by their line spectral frequencies (LSF) [121] optionally. This results in a reduced dimensionality for the feature vector from 256 dimensions for the STFT to 16 dimensions for the LSF representation. Since this step can be regarded as optimization step and has already been applied in our previous work [110], it will not be considered further in the experiments.

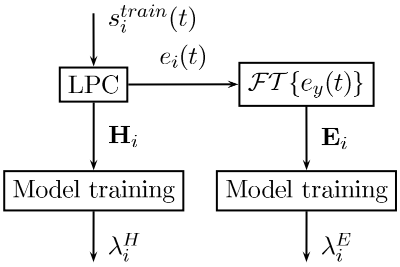


Figure 6.4: Block diagram of the training stage of the source-filter model.

The SCSS decoder structure is shown in Figure 6.5. On the top branch on the left side the spectral whitening of the speech mixture $y(t)$ using LPC is performed. Afterwards, the separation is carried out in two steps. First, the remaining spectral fine-structure $\mathbf{E}_y$ in conjunction with the trained excitation models λ_i^{E} are utilized as input for the *Excitation Separation* unit. This unit decodes the excitation mixture $\mathbf{E}_y$ and extracts the individual excitation signals $\hat{\mathbf{E}}_i$. After the most probable state sequence associated to each speakers excitation has been determined, the means of the emission densities are employed as prototype excitation signals $\hat{\mathbf{E}}_i$. Given the mixture $\mathbf{Y}$ and estimated excitation signals, the VTF models λ_i^{H} of each speaker are employed to adapt the model to the utterance at hand, using Eq.(6.3). Thus, in the *VTF Separation* unit the best fitting vocal tract information is selected from λ_i^{H} for a particular instance of time in the l_2-norm or by the MAP estimate. The provided output $\hat{\mathbf{S}}_i$ is an estimate of the underlying signals.

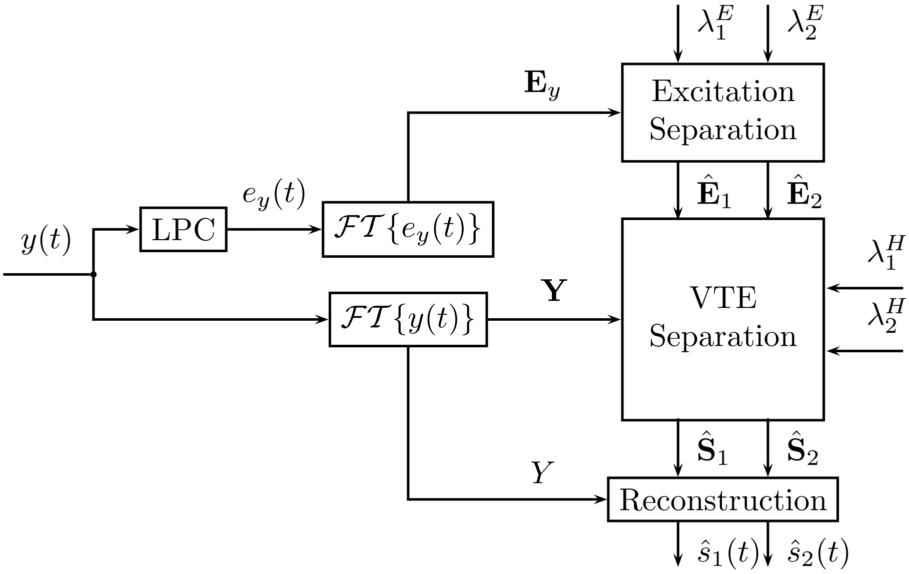


Figure 6.5: Block diagram of the separation algorithm.

Note, in a previous publication [110] we proposed a similar system. In this work, the reported results were inferior due to the limited capabilities of the GMTK toolbox [133]. In the current version of the system, we replaced the toolbox by our own code and, therefore, we present new results.

Source signal representation - Excitation separation

To train excitation models, vocal-fold related signals are extracted for each speaker. Markov chains have been employed to model time dependency between two consecutive states and hence avoid permutations between speakers. For excitation tracking, either learned transition probabilities using training data or a uniformly distributed transition probability are employed. In the latter case, the FM-HMM reduces to a FM-VQ.

In the specific case of the excitation model, the posterior probability given the speech mixture and the FM-HMM is given as $p(u_1, u_2|\mathbf{E}_y) \propto p(\mathbf{E}_y|u_1, u_2) \cdot p(u_1) \cdot p(u_2)$, where $p(u_i)$ are the independent priors of λ_i^{E} and $p(\mathbf{E}_y|\{u\})$ is the likelihood function defined as:

$$p(\mathbf{E}_y|u_1, u_2) = \mathcal{N}(\mathbf{E}_y|\max(\mu_e(u_1), \mu_e(u_2)), \Sigma), \tag{6.11}$$

where $\mathcal{N}$ denotes the normal density, max is the element-wise maximum operator, u_i are the state variables associated to a particular mean $\mu_e(u_i)$, and Σ is the diagonally assumed covariance matrix shared by all speakers. Introducing time-dependency, the best fitting state for each source is extracted according to:

$$\{u_1^\star, u_2^\star\} = \underset{u_1,u_2}{\operatorname{argmax}} \Big[p\big(\mathbf{E}_y | u_1^{(\tau)}, u_2^{(\tau)}\big) p\big(u_1^{(\tau)} | u_1^{(\tau-1)}\big)\ p\big(u_2^{(\tau)} | u_2^{(\tau-1)}\big)\Big], \tag{6.12}$$

where τ denotes time segments, $u_i^{(\tau)}$ the state index for a particular instant of time and $\{u_i^\star\}$ the most probable state of source i given the current observation $\mathbf{E}_y$ and the respective transition probability. Finally, the best sequence can be found using the Viterbi algorithm (see chapter 3, section 3.2.1) and the means of the emission density $\phi_e(u_i^\star)$ of the active state is considered to be an estimate for the fine spectral structure of each speaker. Since every state is modeled by a single Gaussian, the excitation signal is approximated by the means of the emission densities $\hat{\mathbf{E}} = \mu_e(u_i^\star)$.

Envelope modeling - VTF separation

The estimated excitation signals $\hat{\mathbf{E}}_i$ of each speaker as well as the VTF models λ_i^{H} are used to construct UD models for a given observation. Afterwards, the UD model is used for source separation. Thus, this unit delivers the unmixed speech signals. We obtain the most likely state sequence either by the MAP estimate similar to Eq. (6.12), or by the minimization of the l_2-norm as:

$$[v_1^\star, v_2^\star] = \underset{\{v_i\}}{\operatorname{argmin}} \Big|\Big|\ \mathbf{y} - \sum_i \hat{\mathbf{e}}_i\ \cdot\ \lambda_i^{\mathrm{H}}(v_i)\ \Big|\Big|_2, \tag{6.13}$$

where $v_i^\star$ denotes the most likely state. In contrast to the FM-HMM, we do not model any time dependencies. Alternatively, a FM-HMM can be used for VTF separation and the most likely state sequence can be found using Eq. (3.3). For the observation likelihood computation, the state emission density $\phi_s(z_i)$, associated to the random variable z_i is replaced by $\phi_s(z_i) = \hat{\mathbf{E}}_i + \phi_h(v_i)$. The VTF information related to each state $v_i^\star$ at every time step can be combined with the estimated excitation signal and the mixed phase to form an estimate of the component speech signals:

$$\hat{\mathbf{s}}_i = \mathcal{FT}^{-1}\{\hat{\mathbf{e}}_i \cdot \hat{\mathbf{h}}_i \cdot \exp j\angle\mathbf{y}\}, \tag{6.14}$$

where $\hat{\mathbf{h}}_i = \phi_h(v_i) = \lambda_i^{\mathrm{H}}(v_i^\star)$ is the estimated VTF. This signal is either used directly as signal estimate or employed for mask estimation. More details on signal reconstruction can by found in chapter 3, section 3.4. Once the underlying signals are estimated, the speech output sequence is built by the application of the inverse Fourier transform followed by the overlap-add method.

Validation

In the previous sections all system modules as well as their dependencies have been introduced. However, the relationship between the spectrally whitened mixed signal $\mathbf{E}_y$ and the individual LPC residual signals $\mathbf{E}_i$ needs further elaboration. Although there is no closed form relationship between these quantities, we will explore a reasonable approximation. As defined in Eq. (3.9) the two component speech signals are additively related in the time domain. This additivity still holds in the Fourier domain, assuming that the phase information is included. Depicting the signals with their magnitude and phase values, this relation is given as:

$$\mathbf{y}^2 = \mathbf{s}_1^2 + \mathbf{s}_2^2 + 2 \cdot \mathbf{s}_1\ \mathbf{s}_2\ \cos(\psi), \tag{6.15}$$

where ψ is the phase difference between S_1 and S_2. Recently, Radfar et al. [28] have shown that the expected value over the logarithm of Eq. (6.15) results in the max-approximation, i.e. $\mathbf{Y} = \max(\mathbf{S}_1, \mathbf{S}_2)$, assuming a uniformly distributed phase between $[0, \ldots, \pi]$. It should be emphasized that the bin is exclusively assigned to the first source, if source one exhibits more energy in a specific time-frequency bin compared to source two and vice-versa, i.e. the *mixmax*-approach [28]. The derivation in [28] is independent of any signal characteristics. Hence, it is valid also for the excitation signals:

$$\mathbf{E}_{s_1,s_2} = \max\left(\mathbf{E}_1, \mathbf{E}_2\ \right), \tag{6.16}$$

where $\mathbf{E}_{s_1,s_2}$ is the logarithmic frequency representation of the sum of the respective vocal fold excitations in the time domain, i.e., $e_{s1,s2}(t) = e_1(t) + e_2(t)$. Thus, the only relation to show is the validation of $e_{s_1,s_2}(t) \approx e_y(t)$, where $e_y(t)$ is the spectrally whitened speech mixture. As there is no direct analytic relation between these excitation signals, we provide an experimental validation. In the experiment, we combine two utterances of the same speaker at equal level (TMR = $0\ dB$), which is known to be the most challenging case in SCSS. We perform spectral whitening of this speech mixture to obtain e_y. Moreover, we use the excitation signals of the individual speech signals to obtain the mixture of these excitation signals $e_{s1,s2}(t)$. Note, the excitation signals are combined at equal level. We assess performance by computing the mean segmental signal-to-noise-ratio (SNR_{seq}) over the whole speech utterance. We achieve an SNR_{seq} of over 16 dB, where the SNR_{seq} is measured in the log-frequency domain. This is a fairly good value and the approximation supports the assumption. The SNR_{seq} is defined as follows:

$$\text{SNR}_{seq} = \frac{1}{T}\sum_{\tau=1}^{T} 10\log_{10}\frac{\sum_d \mathbf{S}_i(\tau,d)^2}{\sum_d(\mathbf{S}_i(\tau,d) - \hat{\mathbf{S}}_i(\tau,d))^2},$$

where d denotes the frequency bin index. Figure 6.6 shows the mixture of excitation signals $e_{s1,s2}(t)$, the mixture found by spectral whitening $e_y(t)$ and the error signal defined as the difference between the two signals. The SNR_{seq} error of 16 dB, results in an TMR of 8 dB in the time domain, shown at the bottom of Figure 6.6. For better visual inspection, an enlarged section is shown on the right hand side of Figure 6.6.

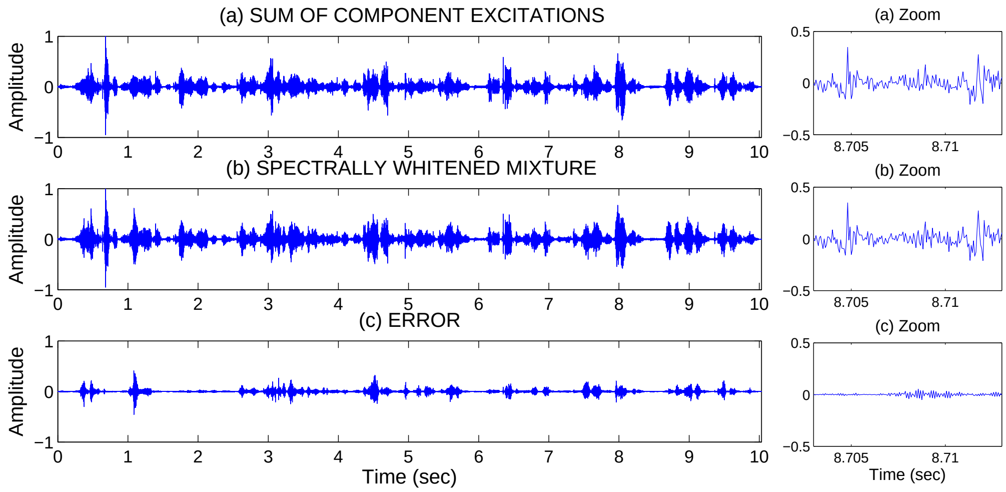


Figure 6.6: (a) Mixture of excitation signals $e_{s1,s2}(t)$, (b) Spectral whitened excitation mixture $e_y(t)$, (c) Corresponding error defined as difference signal: $e_{s1,s2}(t) - e_y(t)$.

Sequential gain-shape source-filter model

The sequential system with statistical excitation estimation can be extended to the sequential gain-shape source-filter model. Therefore, additional gain estimation can be introduced in the above model. This new source separation model is equivalent to the source-filter representation of a speech signal, as shown in Figure 6.1. Prior to model training, the segmental gain information is removed from the speech segments. Thus, all excitation and envelope speech segments are normalized to unit gain and statistical models represent the remaining source and filter shapes. Principally, the same structure as for the system without gain estimation is employed for decoding. Additionally, the gain is estimated, using the ML-based method proposed in chapter 5, section 5.2.1. The independent gain estimation structure of chapter 5, section 5.3.3, as shown in Figure 5.4 is employed for separation. In this structure, for every state of the speaker specific model, the gain is estimated. Afterwards, new state dependent emission densities are defined incorporating the gain information. This step is carried out for every instant of time for both the excitation related model and the VTF model. For the given state sequence of the excitation related model, the estimated excitation signals are synthesized by the addition of the logarithmic gain and the means of the emission densities as:

$$\hat{\mathbf{E}}_i^{(\tau)} = \mu_e(u_i^{(\tau)}) + \log a_e(u_i^{(\tau)}).$$

Once the excitation signals are estimated, the corresponding VTF models are combined to form utterance dependent models, using Eq. (6.3). Thus, the state emission density means are updated at every time step as for the excitation model to $\mu_s(z_i) = \hat{\mathbf{E}}_i^{(\tau)} + a_h(v_i^{(\tau)}) + \mu_h(v_i^{(\tau)})$. Note, in this model, the gain information, as introduced in Eq. 6.2 is split into an excitation a_e and a filter a_h related gain. The most likely state sequence of the new model is employed to find the masks which are finally used for signal separation.

The computational complexity for the statistical source-filter based algorithm can be easily determined, similarly to chapter 4. For the memoryless source-filter based system, two FM-VQ models using the ICM4 algorithm are cascaded. This system possesses the lowest computational complexity of $\mathcal{O}(2 \cdot (T\ M\ 8\ Q))$ with respect to all possible model combinations, i.e. with memory or memoryless models for excitation or envelope separation. In contrast, if for both factorial models a FM-HMM is employed, the computational complexity increases to $\mathcal{O}(4\ TQ\ (4\ M + Q))$. The first part of this complexity originates from the observation likelihood computation, which corresponds to $\mathcal{O}(2 \cdot (T\ M\ 8\ Q))$ for the ICM4. The second part of $\mathcal{O}(4\ T\ Q^2)$ of the computational complexity arises from the Viterbi algorithm used for tracking. M corresponds to the number of iterations the ICM4 takes until convergence. A value of $M = 3$ is selected, analog to chapter 4, section 4.5. This goes a line well with experiments.

6.3 Parallel Source-Filter based SCSS

In the previous section, the source and filter related parts of the speech signals have been estimated in a sequential way. Therefore, the posterior of the model related to the excitation signal has been maximized given the speech mixture. Afterwards, the estimated excitation signals have been used to find the vocal-tract filters of the speech signal. Finally, the combination of the state means have been employed for signal separation. We assumed that this sequential posterior likelihood maximization is similar to the joint posterior likelihood maximization of the source and filter related parts. In this section, we go a step further and perform the joint posterior maximization of the fine- and coarse-spectral structure. Thus, we introduce a model, which estimates the source and filter part in parallel in order to separate a speech mixture. A factor graph representation of this model is depicted in Figure 6.7. In this model each source and filter related part is separately modeled by a Markov chain. The associated hidden random variables are denoted by $u_i^{(\tau)}$ and $v_i^{(\tau)}$, respectively. Thus, two hidden variables each are associated to a speaker with $i \in \{1, 2\}$, where u_i is linked the fine spectral structure and v_i to the coarse spectral structure of the speech signal.

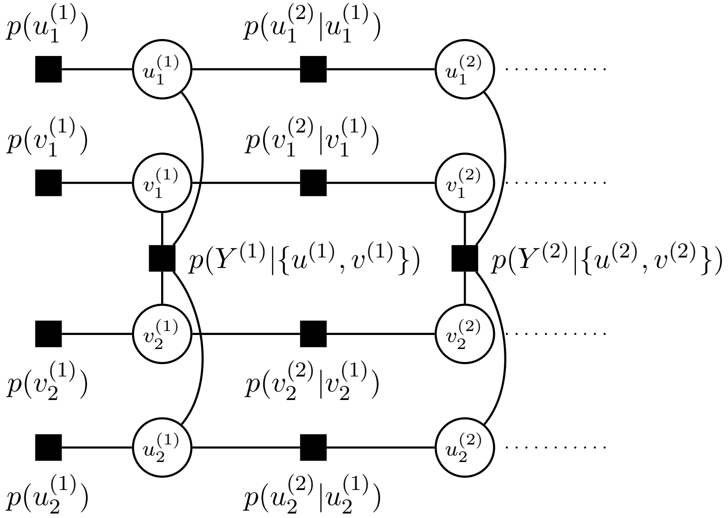


Figure 6.7: Source-Filter F-HMM where $Y^{(\tau)}$ is represented by four state variables $u_1^{(\tau)}$, $v_1^{(\tau)}$, $u_2^{(\tau)}$, and $v_2^{(\tau)}$.

In every hidden chain we define a prior distribution denoted as $p(u_i)$ or $p(v_i)$. In general, the direct conditional dependency between hidden variables are indicated by edges between nodes. The time dependency of hidden variables in each Markov chain is specified by the transition probability $p(u_i^{(\tau)}|u_i^{(\tau-1)})$ and $p(v_i^{(\tau)}|v_i^{(\tau-1)})$. For each hidden chain, we investigate both, the use of dynamics and no dynamics. Therefore, additional transition probabilities are learned from the training data. Note, in our model a direct conditional dependency of u_i and v_i as $p(u_i|v_i)$ is neglected. Moreover, for the transition probabilities, a joint transition, such as $p(u_i^{(\tau)}|u_i^{(\tau-1)}, v_i^{(\tau)})$ is not taken into consideration.

6.3.1 Observation Model

The dependency of the observed variable $Y^{(\tau)}$ on the hidden variables $\{u^{(\tau)}, v^{(\tau)}\}$ for every time instance is defined as $p(Y^{(\tau)}|u_1^{(\tau)}, v_1^{(\tau)}, u_2^{(\tau)}, v_2^{(\tau)}) = p(Y^{(\tau)}|\{u^{(\tau)}, v^{(\tau)}\})$. In our model, the emission density means $\phi_e(u_i)$ or $\phi_h(v_i)$ of each discrete state is regarded as signal prototype. Specifically, each emission density either represents an excitation signal $\phi_e(u_i)$ or the vocal-tract signal $\phi_h(v_i)$. We can use this definition to adapt the observation model of Eq. (6.3) to

$$p(\mathbf{Y}|\{u\}, \{v\}) = \mathcal{N}(\mathbf{Y}; \max(\mu_e(u_1) + \mu_h(v_1), \mu_e(u_2) + \mu_h(v_2)), \Sigma), \qquad (6.17)$$

where $\phi_e(u_i)$ and $\phi_h(v_i)$ form the joint emission density $\phi_s(v_i)$. For simplicity, we omitted the time index in the above equation. The joint observation likelihood computation is a bottleneck in factorial models, as already discussed in chapter 4. In contrast to the previously presented models, in this model, four hidden variables explain the observation. Thus, the computation of the observation likelihood is computationally intractable, i.e. exponentially growing with z, and approximations have to be applied. Specifically, the complexity for four hidden variables is $\mathcal{O}(Q^4\ T)$, where we assume that all random variables have the same cardinality.

Observation likelihood computation: The computational complexity for a single time step of the above model is $\mathcal{O}(Q^4)$. Note, all hidden variable are assumed to have cardinality $Q = |U_i| = |V_i|$, where $u_i \in U_i$ and $v_i \in V_i$. To overcome this problem, the ICM4 method introduced in chapter 4, section 4.5 has been extended to four hidden variables. In order to determine the observation likelihood at the beginning of each time step τ, all random variables are initialized randomly. Afterwards, the likelihood for the selected random variables is calculated and used as upper bound for the algorithm. The random variables connected to one speaker are updated sequentially, for every time step. This means that the update of all random variables of the first speaker is carried out before this procedure is performed for the random variables of the second speaker. The update starts with the variable associated to the fine spectral structure, which is further used for the update of the variable connected to the coarse spectral structure. One iteration step of the ICM4 algorithm is finished by calculating the likelihood, produced by the updated random variables. Finally, this likelihood is compared to that of the previous iteration. If the current likelihood is greater than the previous, another iteration is started, otherwise, the algorithm is stopped. Afterwards, the algorithm is either restarted with a new initialization or a next time step is separated until the end of the speech mixture is reached. The whole observation likelihood computation process for a single time step is summarized in algorithm 6.

```
Input: $\mathbf{Y}$, $\{u\}$, $\{v\}$, $i \in \{1,2\}$
Output: $\{u^m\}$, $\{v^m\}$
Initialize: $\mathrm{LL} = -\infty$, $\mathrm{ll} = 0$, $m = 1$
Randomly initialize: $\{u^m\}$, $\{v^m\}$,
Calculate: $\mathrm{ll} = \log p(\{u^m\}, \{v^m\}|\mathbf{Y})$
Set: $j = 1$ & $k = 2$ or $j = 2$ & $k = 1$
while $ll\ >\ LL$ do
    Set: $\mathrm{LL} = \mathrm{ll}$
    Calculate: $u_j^{m+1} = \mathrm{argmax}_{u_j} \left[p(u_j^m|\mathbf{Y}, u_k^m, \{v^m\})\right]$
    Calculate: $v_j^{m+1} = \mathrm{argmax}_{v_j} \left[p(v_j^m|\mathbf{Y}, u_j^{m+1}, v_k^m, u_k^m)\right]$
    Calculate: $u_k^{m+1} = \mathrm{argmax}_{u_k} \left[p(u_k^m|\mathbf{Y}, u_j^{m+1}, v_j^{m+1}, v_k^m)\right]$
    Calculate: $v_k^{m+1} = \mathrm{argmax}_{v_k} \left[p(v_k^m|\mathbf{Y}, u_j^{m+1}, v_j^{m+1}, u_k^{m+1})\right]$
    Calculate: $\mathrm{ll} = \log p(\{u^{m+1}\}, \{v^{m+1}\}|\mathbf{Y})$
    Set: $m = m + 1$
end
```

Algorithm 6: Iterated conditional modes algorithm for 4 hidden random variables for one time step.

Additionally, this parallel source-filter SCSS model can be extended to estimate the segmental gain associated to each speaker. This results in the parallel gain-shape source-filter SCSS model. Therefore, a gain is introduced for each random variable, separately. Note, this corresponds to a separate gain factor for the excitation related and the vocal-tract filter related parts. A factor graph representation of a single time step is shown in Figure 6.8. Additionally, a random variable linked to each gain is introduced. Therefore, the joint distribution of the observation and source signal, given the source states changes to

$$p(\mathbf{Y}, \mathbf{S}_i|\{u\}, \{v\}, \{a_u\}, \{a_v\}) = p(Y|\mathbf{S}_i)\ p(\mathbf{E}_i|\{u\})\ p(\mathbf{H}_i|\{v\})\ p(a_i|\{a_u\}, \{a_v\}), \tag{6.18}$$

where the relation of $\mathbf{S}_i, \mathbf{E}_i, \mathbf{H}_i$ and a_i is specified in Eq. (6.2). For inference and speech signal reconstruction, the likelihood of the observed mixture is computed as

$$p(\mathbf{Y}|\{u\},\{v\},\{a_u\},\{a_v\}) = \int_{\mathbf{S}_i} p(\mathbf{Y},\mathbf{S}_i|\{u\},\{v\},\{a_u\},\{a_v\})\ \mathrm{d}\mathbf{S}_i. \tag{6.19}$$

As a consequence, the observation model has to be adapted to

$$p(\mathbf{Y}|\{z_i\}) = \mathcal{N}(\mathbf{Y}; \max(\mu_s(z_1), \mu_s(z_2)), \Sigma), \tag{6.20}$$

where $\mu_s(z_i) = \log(a_{u_i}) + \mu_e(u_i) + \log(a_{v_i}) + \mu_h(v_i)$ is the weighted superposition of the excitation ϕ_e and filter ϕ_h emission density means.

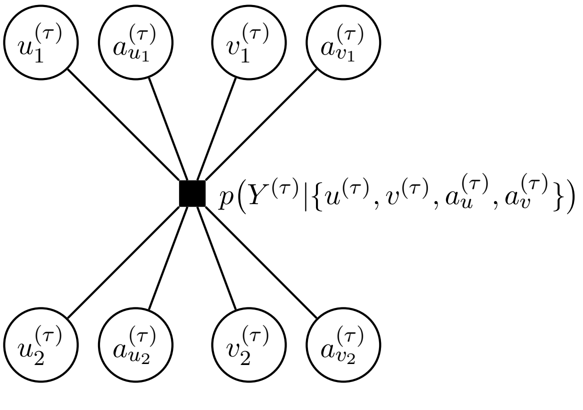


Figure 6.8: Factor graph of the parallel gain-shape source-filter SCSS model, for one time step. Random variable a_{u_i} and a_{v_i} are associated to the introduced gain factor.

The computational complexity of the source-filter model using the ICM4 method for the observation likelihood computation corresponds to $\mathcal{O}(16\ T\ Q\ M)$. Optionally, we represent the parts of the signal with Markov chains. The independence assumption of all random variables, enables the use of a separate Viterbi decoder for each Markov chain for tracking. This results in a complexity of $\mathcal{O}(4\ T\ Q^2)$. Thus, the total computational complexity for the parallel source-filter structure without gain estimation is $\mathcal{O}(4\ TQ\ (4\ M + Q))$. Note, this is the same time complexity as for the sequential source-filter model of section 6.2.2, with two cascaded factorial models.

6.4 Experiments and Results

For the evaluation of the introduced source separation methods, the experimental setup of chapter 1, section 1.5 is used. First, source separation results of the sequential methods are discussed. For the synthetic excitation based method, we use the spectral envelope estimation vocoder (SEEVOC) method described in [131] to split the speech signal into its excitation and VTF related parts. For the remaining methods, we employ linear predictive coding (LPC) [132] to separate a signal into excitation and envelope signals. Therefore, an LPC order of 16 was chosen.

For the sequential method using multi-pitch tracking, differences to the method of Radfar [76] will be elaborated (see also chapter 1, section 1.2.1) and consequences of the additional usage of gain estimation will be discussed. Results for the direct statistical excitation model and the parallel state estimation of all four hidden variables will be reported afterwards. To achieve independency of the separation algorithm concerning the mixing level, gain estimation is included in the algorithms. Finally, performance of the methods are compared to each other with and without gain estimation.

6.4.1 Synthetic Excitation Results

Multi-pitch tracking results

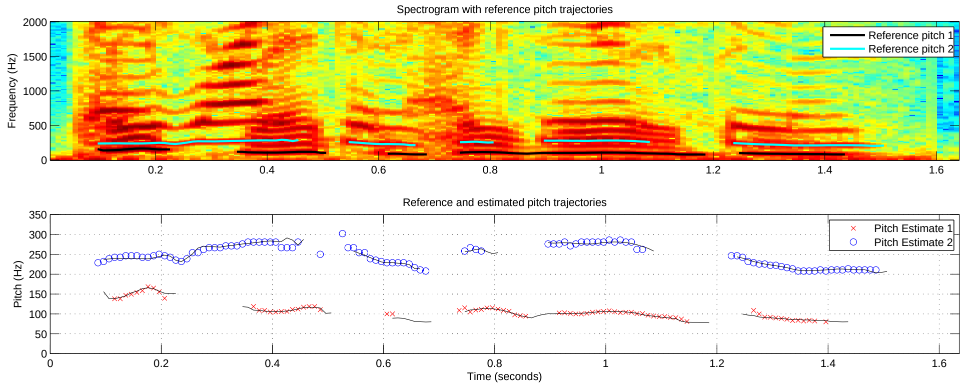
In a previous work [30], we compared the performance of the proposed multi-pitch tracker to the well known approach of Wu et al. [27], and experimentally showed its superior performance on the Mocha-TIMIT database [134]. In the following, we omit any comparisons to other algorithms and report the performance of our approach on the Grid corpus only.
For every test mixture, the method estimates two pitch trajectories, $f_0^1[\tau]$ and $f_0^2[\tau]$. For performance evaluation, each of the two estimated pitch trajectories needs to be assigned to its ground truth trajectory, $\tilde{f}_0^1[\tau]$ or $\tilde{f}_0^2[\tau]$. From the two possible assignments, $(f_0^1 \rightarrow \tilde{f}_0^1, f_0^2 \rightarrow \tilde{f}_0^2)$ or $(f_0^1 \rightarrow \tilde{f}_0^2, f_0^2 \rightarrow \tilde{f}_0^1)$, the one with the smaller overall quadratic error is chosen. Note that this assignment is not done for each individual time frame, but for the global pitch trajectory.
To evaluate the resulting estimates, we use an error measure similar to [27], slightly modified however, to additionally measure the performance in terms of successful speaker assignment. E_{ij} denotes the percentage of time frames, where i pitch points are misclassified as j pitch points, e.g. E_{12} means the percentage of frames with 2 pitch values estimated, whereas only one pitch is present. The pitch frequency deviation is defined as

$$\Delta f_0^i[\tau] = \frac{|f_0^i[\tau] - \tilde{f}_0^i[\tau]|}{\tilde{f}_0^i[\tau]}, \tag{6.21}$$

where $\tilde{f}_0^i[\tau]$ denotes the reference chosen for $f_0^i[\tau]$. For each reference trajectory, we define the corresponding permutation error $E_{\text{Perm}}^i[\tau]$ to be one at time frames, where the voicing decision for both estimates is correct, but the pitch frequency deviation exceeds 20%, and $f_0^i[\tau]$ is within the 20% error bound of the other reference pitch. This indicates a permutation of pitch estimates due to incorrect speaker assignment. The overall permutation error rate E_{Perm} is the percentage of time frames, where either $E_{\text{Perm}}^1[\tau]$ or $E_{\text{Perm}}^2[\tau]$ is one. Next, we define the corresponding gross error $E_{\text{Gross}}^i[\tau]$ for each reference trajectory to be one at time frames, where the voicing decision is correct, but the pitch frequency deviation exceeds 20% and no permutation error is detected. This indicates inaccurate pitch measurements, independent of permutation errors. The overall gross error rate E_{Gross} is the percentage of time frames where either $E_{\text{Gross}}^1[\tau]$ or $E_{\text{Gross}}^2[\tau]$ is one. Finally, the fine detection error E_{Fine}^i is the average frequency deviation in percent at time frames where $\Delta f_0^i[\tau]$ is smaller than 20%. The overall error E_{Total} is defined as the sum of all error terms:

$$\begin{aligned} E_{\text{Total}} &= E_{01} + E_{02} + E_{10} + E_{12} + E_{20} + E_{21} \\ &+ E_{\text{Gross}} + E_{\text{Fine}} + E_{\text{Perm}} \end{aligned} \tag{6.22}$$

where $E_{\text{Fine}} = E_{\text{Fine}}^1 + E_{\text{Fine}}^2$. For our SD models, we train each transition matrix used in the F-HMM on reference pitch data from the corresponding speaker. Moreover, the GMM-based observation model is trained on mixtures of the two corresponding speakers. Similar to [29], experimental results for our SD models suggested that both studied tracking algorithms –



E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Total}
3.0675	0	7.3620	1.2270	0	15.9509	0	3.5844	31.1918

Figure 6.9: Trajectories found by the proposed multi-pitch tracker, applied on speaker MA1 ("prwkzp") and speaker FE1 ("lwixzs") speaking simultaneously. The overall accuracy is high, yet some parts of the trajectory of speaker 1 can not be tracked successfully. This leads to a high contribution of E_{21} and E_{10} to the overall error. The corresponding error measures on this test instance are shown in the Table at the bottom.

the junction tree algorithm and the max-sum algorithm – obtain solutions with equivalent E_{Total}. For this reason, we use the max-sum algorithm for tracking with SD models, as it is more efficient in terms of computational complexity. Table 6.1 shows the resulting error measure on the test set. To illustrate the performance and its corresponding error measure, we show an exemplary tracking result for the SD model in Figure 6.9.

The GD observation models are trained on 3.3 hours of speech mixtures comprising 10 male-male, male-female, or female-female speakers, respectively. The GD transition matrices are trained on reference pitch data of either male or female speakers. In contrast to the SD model case, we observed that the max-sum algorithm performs worse than the junction tree algorithm for GD models applied to same gender mixtures. In that case, the parameters of the F-HMM are the same in each Markov chain. Moreover, the observation likelihood is symmetric in z_1 and z_2, i.e. $p(\mathbf{Y}^{(t)}|z_1, z_2) = p(\mathbf{Y}^{(t)}|z_2, z_1)$. For this reason, we apply the junction tree algorithm for tracking with GD models. Table 6.2 gives the performance results for this model.

Finally, SI models are trained on 6.5 hours of speech mixtures composed of any combination of the 10 male and 10 female speakers. The transition matrix is trained on reference pitch data from both, male and female speakers. For the same reason as for GD models, we use the junction tree algorithm for tracking with SI models. Table 6.3 shows the performance results.

The careful reader will notice that for the SI model, as well as for the male-male and female-female GD model, both Markov chains of the F-HMM have the same transition matrix. In this case the F-HMM only allows symmetric solutions, i.e. identical pitch trajectories. To prevent this, we add a small amount of noise to create two slightly different transition matrices, for each Markov chain. This heuristic breaks the symmetry in the F-HMM and allows individual trajectories for both speakers.

Table 6.1: Performance of F-HMM-based multi-pitch tracking for speaker dependent (SD) training. Mean and standard deviation (std) of the nine test instances of each speaker pair are shown.

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Perm}	E_{Total}
MA1-MA2	Mean	1.97	0.00	6.26	0.84	3.48	25.99	0.73	5.18	2.80	**47.24**
	Std	1.83	0.00	2.63	1.23	4.05	4.63	0.79	0.91	3.75	5.11
MA1-FE1	Mean	1.65	0.00	4.17	1.10	0.49	15.11	0.35	3.61	0.00	**26.48**
	Std	1.23	0.00	2.19	0.85	0.52	5.16	0.33	0.17	0.00	5.70
MA1-FE2	Mean	0.67	0.00	6.43	0.61	0.88	19.82	1.64	3.09	0.13	**33.27**
	Std	0.54	0.00	2.09	0.75	0.69	5.41	1.67	0.22	0.38	5.87
MA2-FE1	Mean	2.23	0.13	4.52	1.11	0.46	12.91	0.07	3.54	0.00	**24.97**
	Std	1.55	0.38	3.10	0.90	0.56	3.97	0.20	0.51	0.00	4.04
MA2-FE2	Mean	1.56	0.00	4.66	1.04	1.41	19.89	0.88	3.37	0.00	**32.82**
	Std	1.71	0.00	2.05	1.15	1.27	7.68	1.27	0.36	0.00	6.23
FE1-FE2	Mean	1.29	0.00	5.46	0.96	1.06	15.02	0.46	5.19	0.44	**29.88**
	Std	1.07	0.00	2.29	1.10	0.76	5.25	0.49	0.36	0.88	4.87

Table 6.2: Performance of F-HMM-based multi-pitch tracking for gender dependent (GD) training. Mean and standard deviation (std) over the 9 test instances of each speaker pair are shown.

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Perm}	E_{Total}
MA1-MA2	Mean	3.89	0.00	6.84	3.84	1.95	21.59	1.89	8.78	5.80	**54.57**
	Std	2.05	0.00	3.26	3.13	2.30	5.70	1.20	2.57	6.37	12.03
MA1-FE1	Mean	3.96	0.00	4.87	3.29	0.77	18.98	1.26	3.87	1.95	**38.95**
	Std	1.86	0.00	2.82	3.49	0.83	3.68	1.11	0.71	1.53	6.38
MA1-FE2	Mean	2.53	0.00	4.93	3.61	1.25	17.64	1.42	3.76	1.45	**36.58**
	Std	1.86	0.00	2.16	1.96	1.51	4.54	1.10	0.64	2.30	8.79
MA2-FE1	Mean	4.05	0.07	2.75	2.42	0.33	14.39	6.08	4.31	2.46	**36.86**
	Std	2.04	0.20	1.76	1.63	0.42	6.05	5.93	0.89	2.72	8.74
MA2-FE2	Mean	2.38	0.00	4.06	3.01	0.52	14.43	2.09	3.90	0.65	**31.04**
	Std	1.63	0.00	2.19	1.85	0.46	4.48	2.65	0.65	0.93	4.56
FE1-FE2	Mean	3.10	0.00	3.56	4.47	0.40	9.90	1.51	6.40	10.18	**39.51**
	Std	2.12	0.00	1.47	2.07	0.58	3.50	1.28	3.16	4.68	6.40

Speech separation results

All modules discussed for the SCSS system using multi-pitch estimation are used to build the source separation algorithm.

For both, λ_{SI}^{VQ} and λ_{SI}^{NMF}, we have trained models with 500 bases, respectively. The dimension of the bases corresponds to the number of frequency bins used in the spectrogram, i.e. 512. For training, we have used 200 iterations for NMF and 150 iterations for VQ, where we perform experiments with and without gain normalized VTF models.

We conducted different experiments with focus on various parts of the system, presented below:

Table 6.3: Performance of F-HMM-based multi-pitch tracking for speaker independent (SI) training. Mean and standard deviation (std) over the 9 test instances of each speaker pair are shown.

		E_{01}	E_{02}	E_{10}	E_{12}	E_{20}	E_{21}	E_{Gross}	E_{Fine}	E_{Perm}	E_{Total}
MA1-MA2	Mean	4.33	0.13	5.39	5.35	0.97	22.09	2.56	7.53	7.93	**56.28**
	Std	1.45	0.38	2.82	3.83	1.03	3.49	1.25	0.87	6.99	11.07
MA1-FE1	Mean	4.10	0.07	4.78	4.95	0.90	17.92	2.72	4.62	17.64	**57.69**
	Std	0.90	0.21	3.20	3.87	0.71	5.58	1.32	1.17	8.63	13.96
MA1-FE2	Mean	2.92	0.00	4.86	4.97	0.78	17.11	2.39	4.13	24.69	**61.83**
	Std	1.78	0.00	2.78	3.96	1.43	4.38	1.51	0.96	16.33	17.92
MA2-FE1	Mean	3.85	0.00	2.42	4.04	0.46	15.10	6.06	3.76	20.95	**56.65**
	Std	2.17	0.00	1.47	2.62	0.76	5.11	4.01	1.34	9.62	11.15
MA2-FE2	Mean	2.89	0.00	3.75	3.68	0.70	12.82	5.20	4.17	19.03	**52.25**
	Std	2.25	0.00	1.71	2.09	0.68	4.60	2.62	1.13	9.29	11.64
FE1-FE2	Mean	4.08	0.00	2.34	2.00	0.29	12.06	1.76	7.14	8.64	**38.31**
	Std	2.20	0.00	0.84	1.95	0.66	3.57	2.23	4.35	4.91	8.10

1. Source separation experiments are carried out using reference $\tilde{f}_0^i$ trajectories for each speaker. The extraction is done on the single speaker utterances using $RAPT$ [2], before mixing and is called the supervised mode. This is the upper bound of the performance, currently achieved by using our method.

2. SD trained models for multi-pitch f_0^i estimation are utilized to separate the speakers. This method is already unsupervised but presumes to know the speaker identities in advance to select the adequate SD models.

3. A GD multi-pitch tracker has been explored to separate the speech mixture.

4. No prior knowledge is assumed anymore and speaker independent models for both, the f_0 estimation and the VTF estimation are employed for separation.

Note, the same SI VTF model is used, in all four experiments. For each of the four different pitch extraction methods enumerated above, we compared four separation approaches, namely, *Exci*, *NMF*, *GE-ML*, and *ML*, explained hereafter:

- *Exci*: The excitation signals, created from the f_0 trajectories by Eq. (6.9), are used for separation. Therefore, binary mask signals are derived based on the excitation signals and the speech signals are finally recovered by filtering the speech mixture with the respective BM.

- *NMF*: NMF is applied for VTF modeling. Utterance dependent bases are found by the combination of the SI learned VTF bases with the excitation signal.

- *GE-ML*: The VQ approach is used to separate the speech mixture. The speaker dependent model is formed by λ_{UD}^{VQ}, using gain estimation. The training data has been gain normalized prior to SI VTF model training.

- *ML*: The VQ approach without gain estimation is employed to separate the speech mixtures. Therefore, the gain information has not been removed from the data during training of the SI VTF model λ_{SI}^{VQ}. For separation, the gain factor has been set to $g = 1$ in Eq. (6.2).

We report results for both, the estimated component signals $\hat{s}_i$ extracted by applying the respective BM on the speech mixture and the synthesis from the estimated speech bases. Naturally, the synthesized signals have a lower quality, compared to the signals extracted directly using the BMs. Nevertheless, the results are rather instructive. A preliminary listening test indicated a subjectively better intelligibility of the synthesized signals, compared to the BM signals for some utterances. This is mainly due to the musical noise introduced by the on/off switching of the binary mask.

In all Figures the achieved mean value is depicted with a red horizontal line. The methods are identified by the label on the x-axis. Moreover, the standard deviation of the TMR is indicated by the blue box, surrounding the red line. All experiments are split into three classes: SGF, SGM, and DG class.

First, performance of the supervised method, using the f_0 extracted by *RAPT* [2] on the single speech utterances are presented. The results for synthesized signals are depicted in Figure 6.10. Those signals are used to estimate the BM for each speaker. Further, the BMs are applied to the speech mixture in order to recover the signals. The BM results are shown in Figure 6.11.

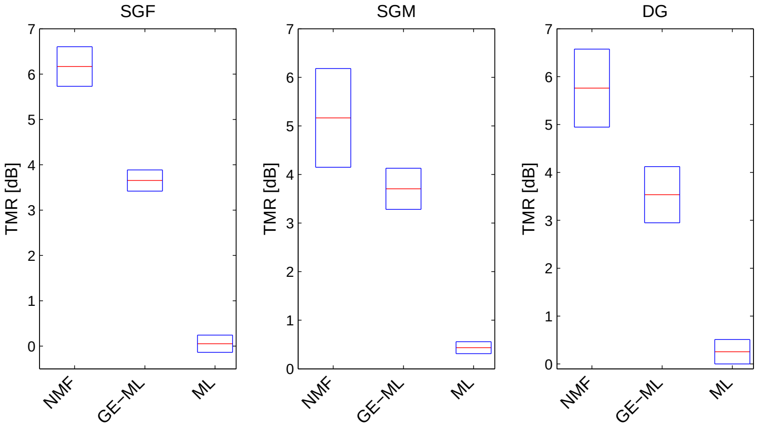


Figure 6.10: Mean and standard deviation of the TMR for the synthesized signals using pitch trajectories extracted by *RAPT* [2].

The performance of *Exci* emphasizes the importance of the fine structure, i.e., harmonics, of speech, which is a major cue for speech separation. This is well known from CASA [3].

Additionally incorporating the VTF models for separation, improves the results in most cases. For the ML based method without gain estimation (GE), the results are getting slightly worse. Surprisingly, for the SGM case the usage of the VTF information does not improve performance at all. We conjecture that the harmonics are rather close to each other and thus, are acting as spikes, which already recover the main speaker specific energy.

Next, the same separation methods are used with SD multi-pitch trajectories to create the excitation signal. In Figure 6.12 the results for the synthesized signals are depicted and Figure 6.13 shows results for the BM signals. As already noted in the above discussion, the separation performance depends strongly on the used fundamental frequency. In our model, the f_0 information introduces at last utterance dependency. Thus, separation performance strongly correlates with the f_0 performance. Nonetheless, the separation results

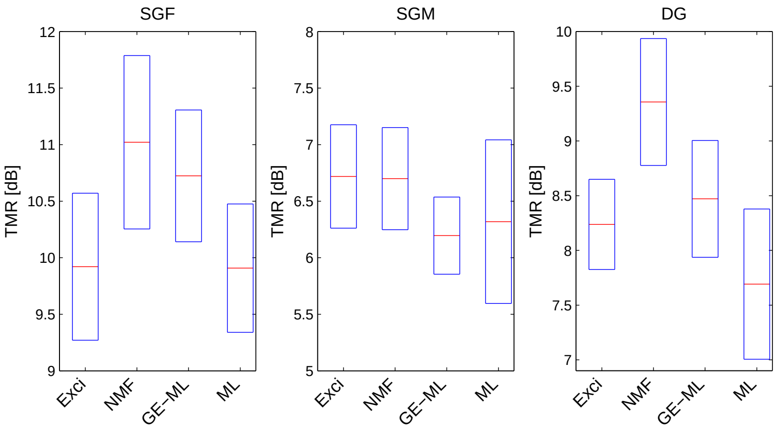


Figure 6.11: Mean and standard deviation of the TMR for the BM signals using pitch trajectories extracted by $RAPT$ [2].

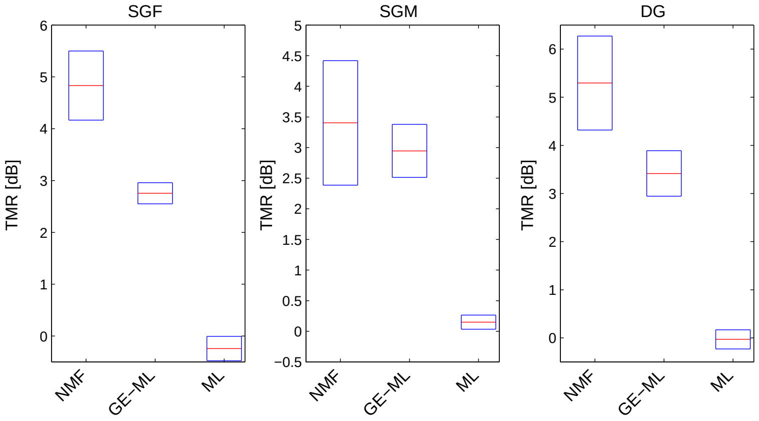


Figure 6.12: Mean and standard deviation of the TMR for the synthesized signals using SD multi-pitch trajectories.

are consistent. The GE-ML method only shows a slightly better performance compared to the excitation *Exci* signal for all cases. Moreover, for the SGM case approximately the same performance for all methods except the ML one can be reported using the BM. Similarities can be drawn to CASA where the separation is carried out in two steps: simultaneous and sequential grouping. In our system, simultaneous grouping is executed during separation and sequential grouping is treated during multi-pitch tracking. In this respect, the sequential grouping is measured by E_{Perm}. For the SD case, Table 6.1 shows that a permutation error occurs rarely. Averagely, 0.03% for different gender and 1.62% for same gender mixtures of the speech frames are permuted.

As an intermediate step towards SI SCSS, gender dependent multi-pitch tracking models to estimate f_0 trajectories are applied. Figure 6.14 shows the results for the synthesized and Figure 6.15 for the BM signals. Here, the same transitions are employed to estimate the pitch trajectories for the SG cases. Different transitions are only taken for the DG case, which leads to a more accurate pitch estimation and consequently, to a better separation performance. Moreover, the permutation error for same and different gender mixtures occur on average

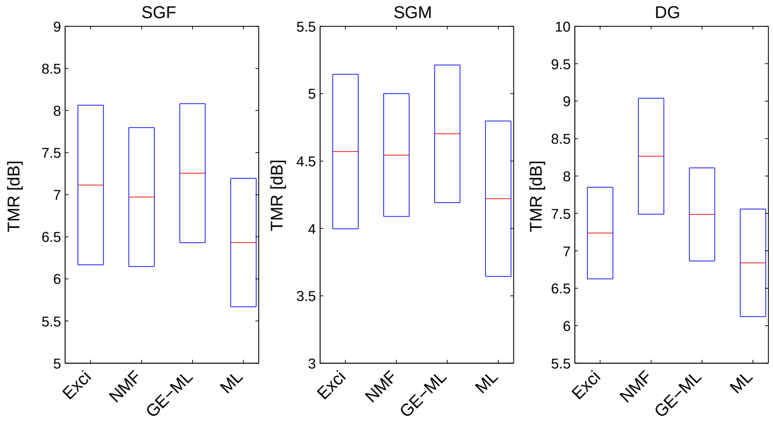


Figure 6.13: Mean and standard deviation of the TMR for the BM signals using SD multi-pitch trajectories.

in 7.99 % and 1.63 % of the speech frames, respectively. Both errors are coherent with the separation results.

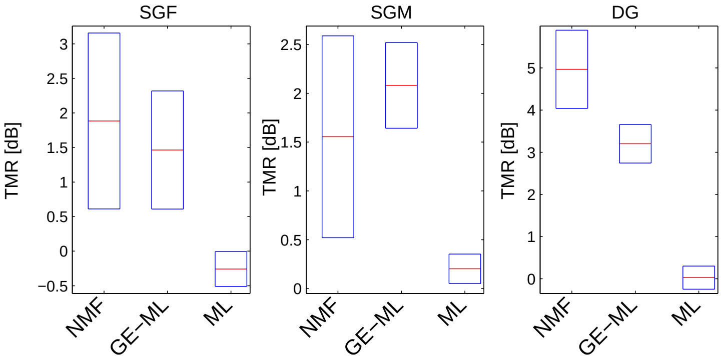


Figure 6.14: Mean and standard deviation of the TMR for the synthesized signals using GD multi-pitch trajectories.

Finally, SI extracted f_0 trajectories are employed for speech separation. This case is a fully SI SCSS method. Again Figure 6.16 and 6.17 show the results for the synthesized and the BM extracted signals, respectively.

For the SI results, the GE-ML method provides slightly better performance, using the synthesized signals. Nonetheless, big differences among the methods could not be found within the investigated BM signals. The synthesized signals of the ML method show a rather poor performance. For different gender mixtures, E_{Perm} increases to 20.58% on average. In contrast, for same gender mixtures E_{Perm} is on average 8.28%. This is about the same E_{Perm} as for the GD models. Thus, for different gender mixtures, sequential grouping is a problem, which is reflected by the significant contribution of E_{Perm} to E_{Total}. This also limits the source separation performance. This issue can be mitigated by post-processing, e.g. Shao et al. [14] recently proposed a clustering approach to perform sequential grouping. In summary, we have shown an almost linear relation between the separation results and the multi-pitch estimation performance when moving from the supervised to the SD and finally to SI based pitch estimation. This is shown in Figure 6.18 (a) and (b), which present the

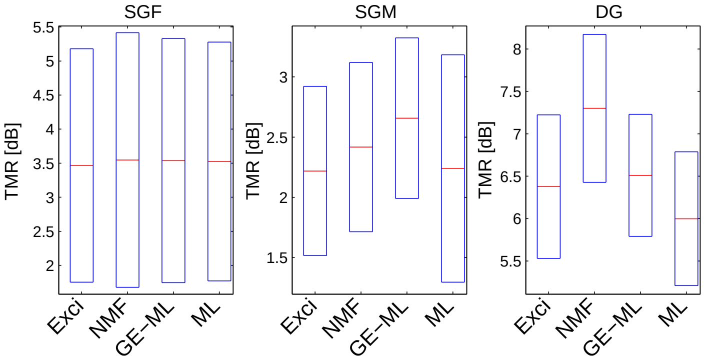


Figure 6.15: Mean and standard deviation of the TMR for the BM signals using GD multi-pitch trajectories.

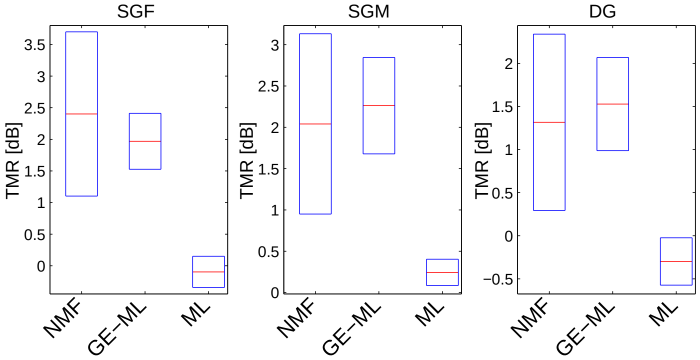


Figure 6.16: Mean and standard deviation of the TMR for the synthesized signals using SI multi-pitch trajectories.

coherence between the TMR for all introduced speech separation methods and the E_{Total} of the pitch tracker. Results are separately depicted for the reference, SD, GD, and SI multi-pitch trajectories. If the E_{Total} is increasing, the TMR is decreasing, no matter, which method is selected for separation. It was already shown in the Figures 6.10, 6.12, 6.14, and 6.16 that the synthesized ML signals are inappropriate to make them directly audible independent on the pitch estimation method. Moreover, it can be seen from Figure 6.18 (b) that the NMF and GE-ML methods show almost the same performance averaged over all pitch extraction models. The ML method leads to a decrease of the TMR performance compared to the BM signals extracted from the excitation (Exci) signals alone. It should be noted that the phonetic content of the utterances was approximately the same except one different word in the sentence (see Table 1.2). In a nutshell, the comparison of all proposed VTF models slightly favors NMF.

The computational complexity of each module has been addressed in the previous sections. The overall complexity of the system is the cumulation of these complexities. The average length of the speech mixtures is 1.69 [sec]. This time is compared to the average time, which is needed to separate an utterance. Therefore, we measure the average time of each system module: The multi-pitch observation likelihood computation and tracking takes on average 862 and 18 [sec], respectively. However, note that the likelihood computation amounts to the evaluation of a set of GMMs, which can be computed in parallel to a high degree. In our

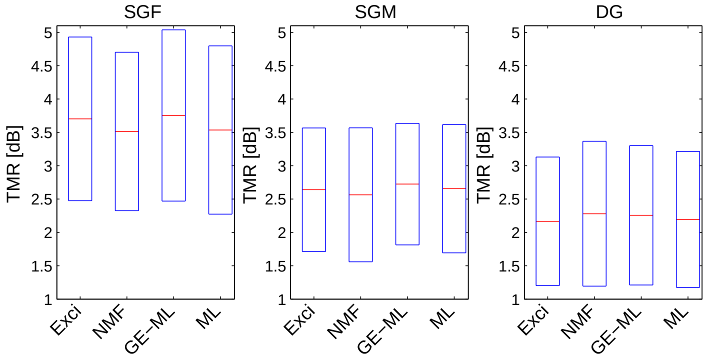
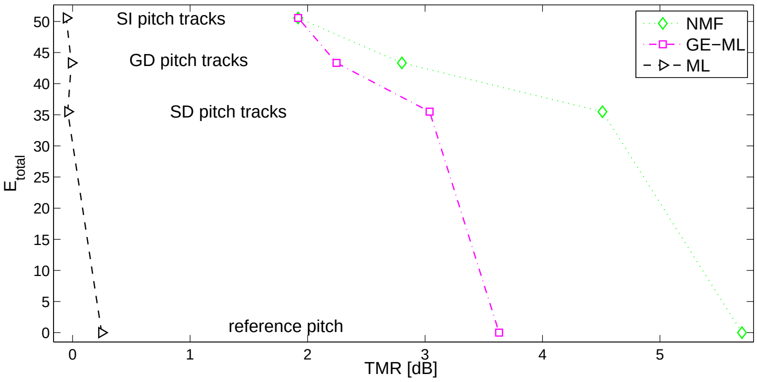
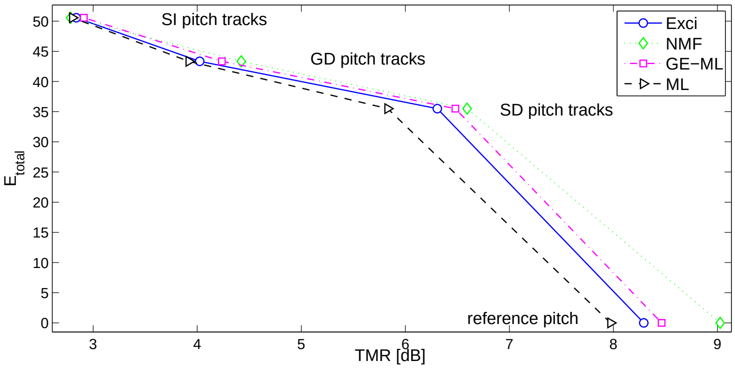


Figure 6.17: Mean and standard deviation of the TMR for the BM signals using SI multi-pitch trajectories.

evaluation, only sequential computations were performed. The VTF observation likelihood calculation using the BS method takes 4.4 [sec] on average. To separate one speech file of average length 1.69 [sec], the system takes approximately 884.4 seconds. Hence, 97.5% of the processing time is currently used for the observation likelihood computation during pitch tracking. All experiments have been performed using MATLAB on an Intel CPU CORE-i7 QUAD 920 running at 2.66GHz. However, computational costs can be further reduced by approximations [108, 107].



(a) E_{Total} versus TMR: Synthesized signals.



(b) E_{Total} versus TMR: BM signals.

Figure 6.18: Coherence between the average E_{Total} versus average TMR for all VTF and pitch estimation methods for the reference, SD, GD, and SI pitch tracks. Results are separately plotted for (a) synthesized speech signals; (b) BM signals.

6.4.2 Separation Results for the Direct Statistical Excitation Model

In this section, the performance for the direct statistical modeling of the excitation signal is discussed. For these experiments, the structure shown in Figure 6.5 of section 6.2 is employed. In the first experiment, the gain is inherently modeled by the excitation and VTF models. In the second experiment, however, gain information is additionally estimated. For the observation likelihood estimation in case of the HMM for both, the excitation and VTF model, the ICM4 algorithm of chapter 4, section 4.5 has been used. For the memoryless models (VQ), we also utilized the ICM4 in case of the excitation related model and the MSE measure to extract the VTF state sequence. In Figure 6.19, separation results are shown for the sequential model without gain estimation. The labels on the x-axis indicate the different structures, employed for separation. For example, the structure with label *HMMVQ* utilizes the FM-HMM for the separation of the excitation related signals. The estimated excitation signals synthesized by the state sequence of the emission density means $\phi_e(u_i)$ are further used to estimate the vocal tract filters using the FM-VQ model. Thus, for the sequential source-filter model, four different structures, with or without dynamics, can be investigated, namely:(i) VQVQ, (ii) VQHMM, (iii) HMMVQ and (iv) HMMHMM.

The labels of Figure 6.19 indicate that all combinations of the introduced structures are studied. A comparison of the various employed structure combinations show no significant differences in performance, measured in TMR. In the SGM case, however, the *HMMVQ* system is favorable. For this source-filter model, the results propose the use of the FM-VQ model for excitation and VTF modeling, due to the decrease in computational complexity.

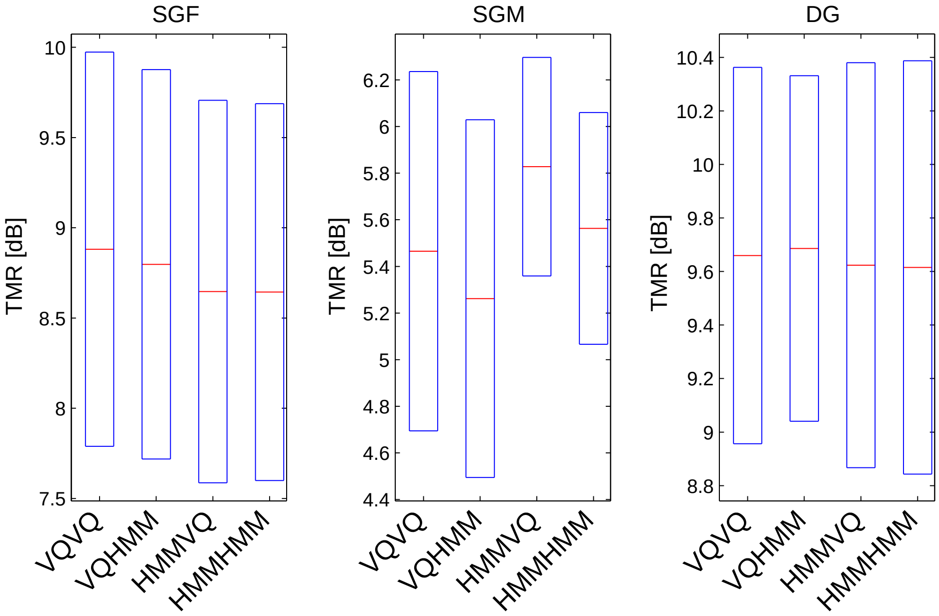


Figure 6.19: Mean and standard deviation of the TMR for the sequential source-filter based SCSS methods. Here different graphical models structures are investigated to estimate the fine- and coarse spectral signal parts.

In the second experiment, the segmental gain information is estimated additionally. This model is equivalent to the source-filter representation of a speech signal, as shown in Figure 6.1. Thus, this model can be applied for arbitrary mixing levels. Since two additional unknown parameters are estimated in this model, we expect a decrease in separation performance for an equal mixing level. The source separation results using gain estimation are shown in Figure 6.20. Again, all four structures have been evaluated in the experiments. For the SGF and DG cases, no significant differences amongst the different structures can be reported. For the SGM case however, noteable differences can be reported, similarly to the first experiment without gain estimation. We notice that there is a steady decrease from memoryless models to models with memory. Specifically, the *HMMHMM* structure results in a significantly lower achieved average TMR. A comparison to the sequential model without gain estimation shows a slightly lower TMR for the SGF but a higher TMR for the DG case. Note, although we additionally estimate two parameters, the performance measured in TMR remains almost constant.

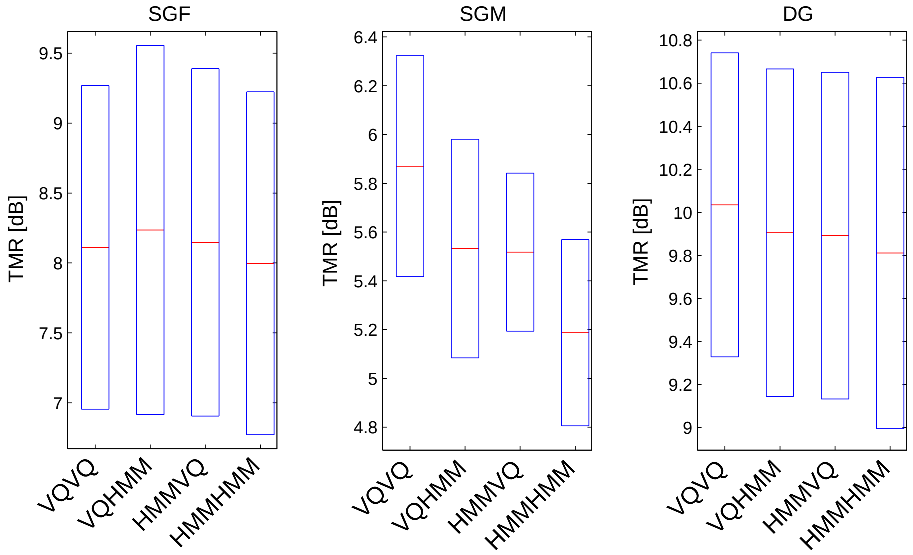


Figure 6.20: Mean and standard deviation of the TMR for the BM signals using sequential gain-shape source-filter based SCSS method. Here different graphical model structures are investigated for gain-shape source-filter based source separation.

Furthermore, we note that the spectrally whitened speech mixture well approximates the underlying excitation signals of each speaker. Thus, the separation results are in accordance with the experimental validation of this approximation (see section 6.2.2).

6.4.3 Parallel Source-Filter Model Separation Results

The introduction of the ICM4 algorithm offers a way to make a factorial model with four parallel evolving Markov chains computationally tractable. This section, discusses the results for both factorial models, without and with gain estimation. For gain estimation, the ML-based method, introduced in chapter 5, section 5.2.1 has been applied. For both experiments, we evaluate all variations of dynamic modeling. In case of the models with dynamics, the system uses the Viterbi decoder to find the most likely joint state sequence $u_i^{1..T}$ or $v_i^{1..T}$ associated to every hidden variable. Moreover, performance is assessed using the specified test set. A cardinality of $Q = |U_i| = |V_i| = 500$ was specified as model size to train all models. The training speech data was separated into its fine- and coarse-spectral structure using an LPC order of 16. We normalize the speech material prior to training in case of the factorial model with gain estimation. Therefore, each time frame in the magnitude frequency domain is normalized to unit norm.

Figure 6.21 depicts the results for the factorial model with four hidden random variables. The labels on the horizontal-axis, specify the different combinations of models with and without dynamics. For all structures, an average TMR increase of $\sim$ 10.5 dB, 6.5 dB and 10.8 dB is achieved for the SGF, SGM and DG case, respectively. We observe, that the additional use of dynamics does not result in an increase of performance. However, the complexity is increased significantly. Moreover, for the SGM case, the use of dynamics for the excitation estimation reduces the performance slightly.

Additionally, we investigated different update schemes for the hidden variables during observation likelihood computation in the ICM4 algorithm. For the conducted experiments, always the fine-spectral structure related to each speaker was updated at the beginning. TMR results for the reversed random variable update scheme, showed no significant differences. Just for the SGM case a decrease of 0.5 dB in TMR was observed.

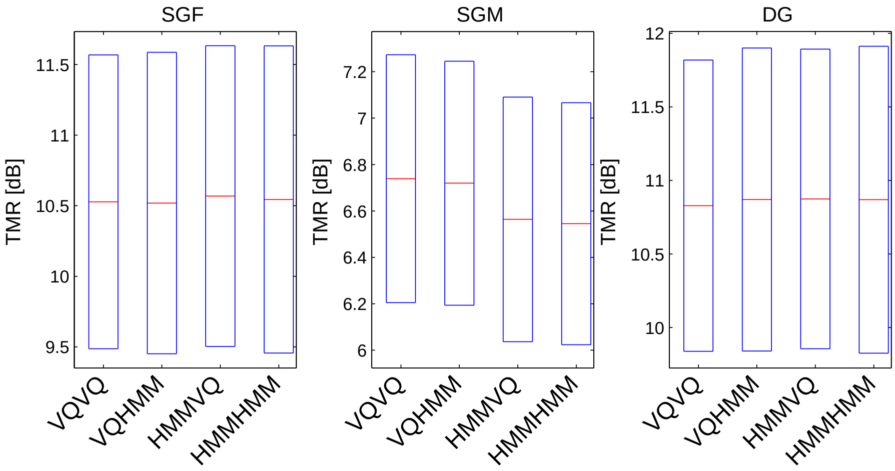


Figure 6.21: Mean and standard deviation of the TMR for the BM signals using the parallel source-filter based SCSS method.

Next, we carry out the same experiments, but this time we additionally employ gain estimation for signal separation. In this case, there are two more degrees of freedom for each speaker. The four random variables $\{a_u\}$ and $\{a_v\}$ account for the gain determination at every time step. The results for the parallel gain-shape source-filter model are shown in Figure 6.22. For all model combinations, a mean TMR increase to 10.3 dB for the SGF, 5.3 dB for the SGM, and 10.5 dB for the DG case can be reported. A comparison to the system without gain estimation results in no significant decrease in performance for the SGF and DG case. However, for the SGM case, a slight drop in performance from approximately 6.5 dB to 5.4 dB has to be reported. Note, the additional use of dynamics for speaker dependent acoustic models does not increase the separability. Further, the additional estimation of the gain associated to each time frame, enables the application of this separation algorithm for arbitrary mixing levels at the cost of a marginally reduced TMR. Additionally, we performed two more experiments with the parallel gain-shape source-filter method. In the first experiment, the method was applied to real world mixtures. The experimental setup as well as results are reported in appendix A. In the second experiment, this method is studied as pre-processing for an automatic speech recognition system. In appendix B we discuss the experimental setup as well as results. Moreover, results are compared to results achieved by clean speech, mixture data, and separated data using the FM-VQ state-of-the-art algorithm.

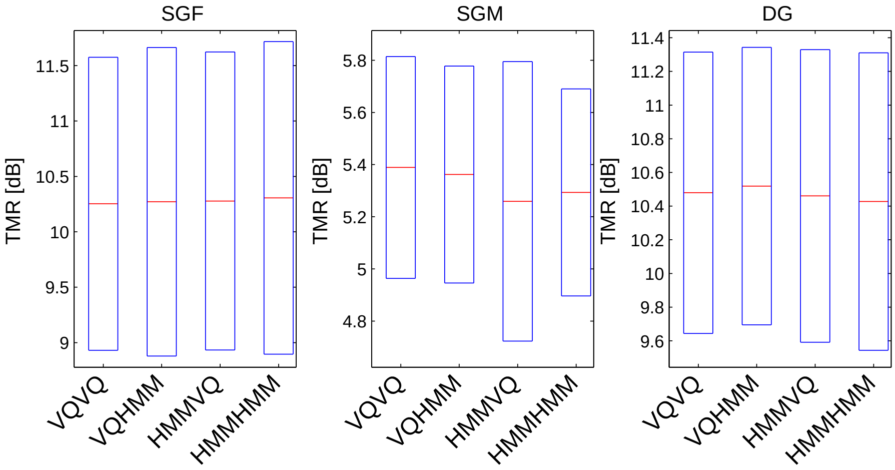


Figure 6.22: Mean and standard deviation of the TMR for the BM signals using parallel gain-shape source-filter based SCSS methods. Here different graphical model structures are investigated for gain-shape source-filter based source separation.

6.4.4 Sequential versus Parallel Source-Filter SCSS

Figure 6.23 summarizes the best results of all proposed systems within this chapter. These are the sequential source-filter system using multi-pitch tracking in the source-driven part (seq-AE), the sequential method with the statistical representation of the source-driven part (seq-VQVQ), and the parallel structure, once using LPC (par-VQVQ-lpc) and once using SEEVOC (par-VQVQ) to find a source and filter representation. The last three listed methods perform decoding without the use of dynamics, as indicated in the labels. Only the multi-pitch unit employs dynamics for tracking. The results emphasize the superior performance of the parallel source-filter structure for all three different mixture cases. Alternatively, we compare the LPC and the SEEVOC [130] method to split the signals into its fine- and coarse-spectral structure for the parallel model. The LPC based method increases the TMR more, as shown in Figure 6.23.

6.5 Conclusion

This chapter discussed the application of the well known source-filter representation of speech signals for single channel source separation (SCSS). The source-filter representation for SCSS has been already discussed by Gomez et al. [35] and Radfar et al. [76]. While Gomez proposed signal separation by the prediction of future frequency cell energy based on current estimated neighboring frequency cells, Radfar united source- and model-driven aspects in order to perform separation. This resulted in a sequential system, which first extracted pitch and afterwards vocal tract related information. The methods discussed in this chapter, rely on this sequential system with multi-pitch estimation. But in contrast to Radfar, a probabilistic factorial HMM was employed for multi-pitch extraction. Moreover, to account for different mixing levels, gain estimation has been discussed and different statistical models for vocal tract modeling have been evaluated. Afterwards, the system was extended, replacing the multi-pitch tracking unit by directly modeling the fine spectral structure. This system is advantageous as also unvoiced speech is explicitly modeled. For the pitch-based system, the discriminative feature is lost during unvoiced segments of both speakers. Finally,

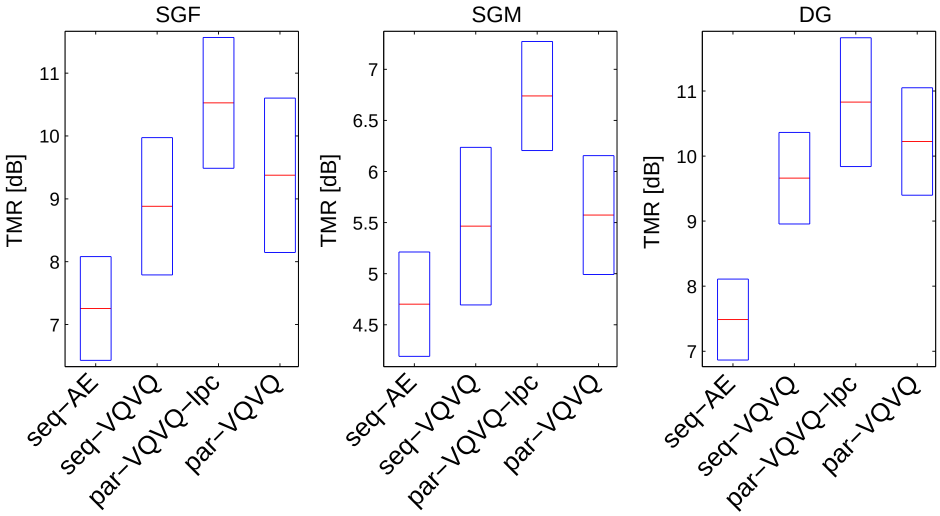


Figure 6.23: Mean and standard deviation of the TMR. Comparison of sequential and parallel source-filter based SCSS algorithms. *seq-AE* and *seq-VQVQ* denote the sequential methods using multi-pitch tracking and the direct excitation signal model. *par-VQVQ-lpc* and *par-VQVQ* refer to the parallel methods using LPC and SEEVOC to find the coarse and fine spectral structure, respectively.

a hidden variable for the fine- and coarse-spectral structure of each speaker was introduced and tracking was performed in parallel. The state sequence of four hidden variables was extracted simultaneously. Since exact inference is computationally intractable, the ICM4 observation likelihood approximation introduced in chapter 4, section 4.5 was incorporated. For all proposed models, gain estimation was included and evaluated. All conducted experiments suggest the use of the parallel source-filter estimation system. From a computational complexity point of view, the sequential model with statistical excitation modeling and the parallel model have the same complexity. Only the model using multi-pitch extraction has a significantly higher computational complexity.

Chapter 7

Conclusion and Future Directions

This chapter starts with a summary on single channel source separation (SCSS) and lists the contributions of this thesis. However, there still remain open questions to be addressed in the future. Consequently, we raise possible research directions towards achieving human performance.

7.1 Summary and Contributions

In SCSS, multiple source signals are extracted from a single observation. This is an underdetermined problem and hence requires further constraints in order to be solved. Statistical independence, constant statistics over time or source prior knowledge can be employed as constraints. Chronologically, the first systems tried to mimic the human auditory system in order to separate signals. This started with the search for the same low level cues which humans use as well as the effort to model the neural activities in the human brain. Recently, machine learning algorithms emerged in the field of SCSS and showed its potential for this task. These methods explicitly model source specific characteristics. Due to the increased availability of computational power and memory storage, these methods are achieving great importance also for online processing systems. However, these systems suffer from three problems: (i) huge computational complexity in factorial models. (ii) the models match only for equal mixing level and (iii) the speaker identities must be known a priori.

Chapter 1 introduced SCSS and partial shortcomings of current systems. Moreover, efforts were made to compare this thesis to related theses and to emphasize its specific contribution. Chapter 2 introduced an overview of methods in SCSS and reviewed methods to assess separation quality. However, the measure of separation quality strongly depends on the target application. The performance for the developed algorithms was assessed using a simple signal-to-noise measure, which compares the energy of the reference to the separated signal. We referred to this measure as target-to-masker-ratio to emphasize the masking aspects of the interference. Additional low-level objective measures which also account for artificial energy and interference energy were introduced by Vincent et al. [80]. Nevertheless, for simple comparisons a singular valued measure seems to be preferable.

The beginning of chapter 3 presented a summary of model-based methods and analyzed each building block in detail. Based on this knowledge, we addressed the issue of model complexity and model quality after training, and we compared performance to a simple template-based system. For signal reconstruction, the construction of the *soft binary* mask was proposed. An automatic speech recognition (ASR) experiment on the *SAIL real-life corpus* [21], compared the three different reconstruction masks. In this experiment, the application of a *soft binary* mask resulted in a lower word error rate compared to the two other masks (see appendix B).

Due to the factorial nature of model-based systems, chapter 4 discussed efficient ways to decrease computational complexity. The main complexity originated from the computation of the observation likelihood during inference. Therefore, we described efficient methods for decreasing the complexity by up to two orders of magnitude while only marginally reducing the separation performance.

Model-based systems perform best at equal mixing level, which is limiting its application. Chapter 5 presented various strategies to estimate the gain for each speech segment individually. This approach led to the shape-gain factorial model for speech separation. We argued to determine the gain on a segment basis instead of estimating the mixing level for a whole utterance as suggested in [5, 24]. The shape-gain representation resulted in three different structures, namely, the shape-gain, the gain-shape, and the independent gain-shape decoder structures, to determine the best fitting shape and gain for each speaker. Interestingly, the independent shape-gain estimation structure showed superior performance. A comparison of the six proposed gain estimation methods favored the maximum-likelihood based gain estimation.

In chapter 6, attributes of implicit models had been integrated in the explicit model. This led to the representation of the speech signal by its fine- and coarse- spectral structure, in analogy to the source-filter representation of speech. As a first step, the comb-like structure of the excitation signal in the frequency domain was modeled by the fundamental frequency, whereas a memoryless model (Vector Quantizer) captured the characteristics of the shaping of the vocal tract. Multi-pitch tracking was performed, using a probabilistic approach. Afterwards, the pitch information was combined with the vocal tract model, which yielded in utterance dependent models. Separation was achieved combining these utterance dependent models into a factorial model. The utterances of the Grid corpus [20] contained mainly voiced utterances, therefore, the separation performance was reasonably high. However, the discriminative feature is lost for this method during unvoiced sections of both speakers. As a consequence, the direct representation of the excitation related signal was introduced. In this step, we kept the sequential structure of the system for separation. Since the excitation signal was modeled directly, the speech mixture could not be employed directly for separation. Therefore, we proposed the spectrally whitened speech mixture as approximation of the sum of the individual excitation signals. Finally, the parallel estimation of the source and filter signals is discussed. This led to a factorial model with four hidden discrete random variables. The application of the iterated conditional modes observation likelihood approximation of chapter 4 enabled computational feasibility. The analysis of the systems in terms of computational complexity showed an equal complexity for the sequential and parallel structure with direct excitation modeling and a significantly higher complexity for the separation model with multi-pitch estimation. Note that the use of the parallel structures makes the spectrally whitened speech mixture a redundant observation. Additionally, gain estimation was integrated into all three proposed systems. This resulted in the so-called gain-shape source-filter SCSS system, which corresponds to the production model of speech [23]. We can report an increased separation performance for all but the same gender male mixtures compared to the source-filter model without gain estimation, although the source-filter shape-gain model estimated two additional parameters.

A last experiment focused on the application of the parallel source-filter shape-gain model in a real cocktail party environment (see appendix A). Therefore, the single speaker utterances of two speakers had been played back by two loudspeakers and recorded with an omni-directional microphone. For different distance settings between loudspeakers and microphone, the separation performance is reported. At the current stage, the model can not deal with convolutive mixtures, thus reverberation is not removed. We can report an average TMR increase of 7.23 dB and 6.43 dB for distances of 50 cm and 1 m between microphone and loudspeakers, respectively. Finally, we evaluated the proposed method in terms of the word error rate on artificial speech mixtures of the WSJ0 database [38]. For the automatic speech recognition application the WER was reduce from 106.9 % for the speech mixture to 73 %.

To put it in a nutshell, this thesis successfully applied the full standard speech production model to single-channel source separation. Each module was evaluated and compared to other work. Moreover, solutions were provided to overcome the computational complexity problem in factorial models. The final system was evaluated with linear instantaneous as well as real speech mixtures via the TMR and the WER.

7.2 Future Directions

This thesis addressed some problems in single channel or monaural source separation. However, it is still far from achieving human performance on a wide range of possible interferences and different environmental conditions. As an example, human performance was exceeded the first time by Kristjansson et al. [5], on a restricted recognition task.

This thesis was restricted to separate co-channel speech, where the interference is known to be speech. In the open-set case of unknown interferences, as already discussed in the introduction, the separation is much more challenging. To develop one algorithm that can deal with all interferences seems to be sheer impossible. It is more likely to use a whole bunch of algorithms, each for a specific task or class of interference and apply them appropriately. For this approach, however, reliable classification algorithms are needed.

Another open issue is the application of single channel source separation algorithms in real environments. Currently, a linear instantaneous mixture model is assumed, which will not hold for convolutive mixtures. Thus, there is no way around to develop convolutive mixture models for single channel methods as proposed for multi-channel algorithms [4, 135].

A major shortcoming of the system of this thesis as well as of most state-of-the-art separation algorithms is the assumption of *a priori* known speaker identities. Despite the fact that in the model of Kristjansson et al. [5] speakers are selected from a set automatically, the application to an open-set of speakers is questionable. Instead, a future direction might employ information theoretic approaches to determine source dependent clusters online or to find a strategy, which introduces source dependency in an iterative manner. Current systems mainly rely on the speaker identity as discriminative feature for separation. This feature breaks the symmetry in factorial models and enables speech separation. The integration of high level features for separation could make the speaker identity a redundant feature. Specifically, a speaker independent trained phone model could trigger the vocal tract model of the proposed source-filter model, and thus break the symmetry of factorial models. Such models might supersede speaker dependent models.

Appendix A

Experiments in a Real-World Environment

The parallel source-filter shape-gain separation algorithm proposed in chapter 6, section 6.3 has been developed for co-channel speech. In this chapter we apply this algorithm to convolutive speech mixtures recorded in a real environment. To the best knowledge of the author, this case has not yet been studied in the literature for single channel source separation (SCSS). At the beginning, the recording setup is introduced. Besides, the geometry of the room as well as the relative position of loudspeakers and microphone are described. The recording equipment is defined and finally results are presented. We draw comparisons to the multi-channel work on convolutive mixtures in the discussion.

A.1 Recording Setup

In all previous experiments, the speech mixture was assumed to be co-channel speech. In this experiment, the proposed parallel gain-shape source-filter single channel source separation approach is applied to data of a real environment. Therefore, the respective source signals of the specified test data, taken from the Grid corpus [20] are played back by two loudspeakers. The propagating signals are interacting with each other and the room. The resulting mixture is captured at the microphone location. Additionally, each source signal was played back and recorded separately for reference. Thus, alltogether 54 speech mixtures are recorded under the influence of the room acoustics. The recordings have been carried out in the Cocktail Party Room (CPR) of the *Signal Processing and Speech Communication Laboratory* at Graz University of Technology. The ground plan is shown in Figure A.1. The CPR room has the dimensions (length × width × height) of 5.72m × 5.32m × 4.14m. The loudspeakers as well as the microphone were placed at the same height of 1.25m from the floor. Moreover, the loudspeakers and microphone are placed in the room asymmetrically, i.e. the distances to each wall are different. The distance $d1$ between each loudspeaker and the microphone is the same. $d2$ denotes the distance between the two loudspeakers. Alltogether 3 experiments with different setups for $d1$ and $d2$, specified in table A.1, were performed. An equivalent network of the recording setup is shown in Figure A.2. The different transfer functions between loudspeakers and microphone are modeled by h_1 and h_2, respectively. Thus, the speech mixture changes from a linear instantaneous model $y(t) = s_1(t) + s_2(t)$ to

$$y(t) = s_1(t) \star h_1 + s_2(t) \star h_2,$$

where $\star$ denotes the convolution operator. Note, in this model we assume time-invariant impulse responses, which is in general not true. For our experiment this assumption goes

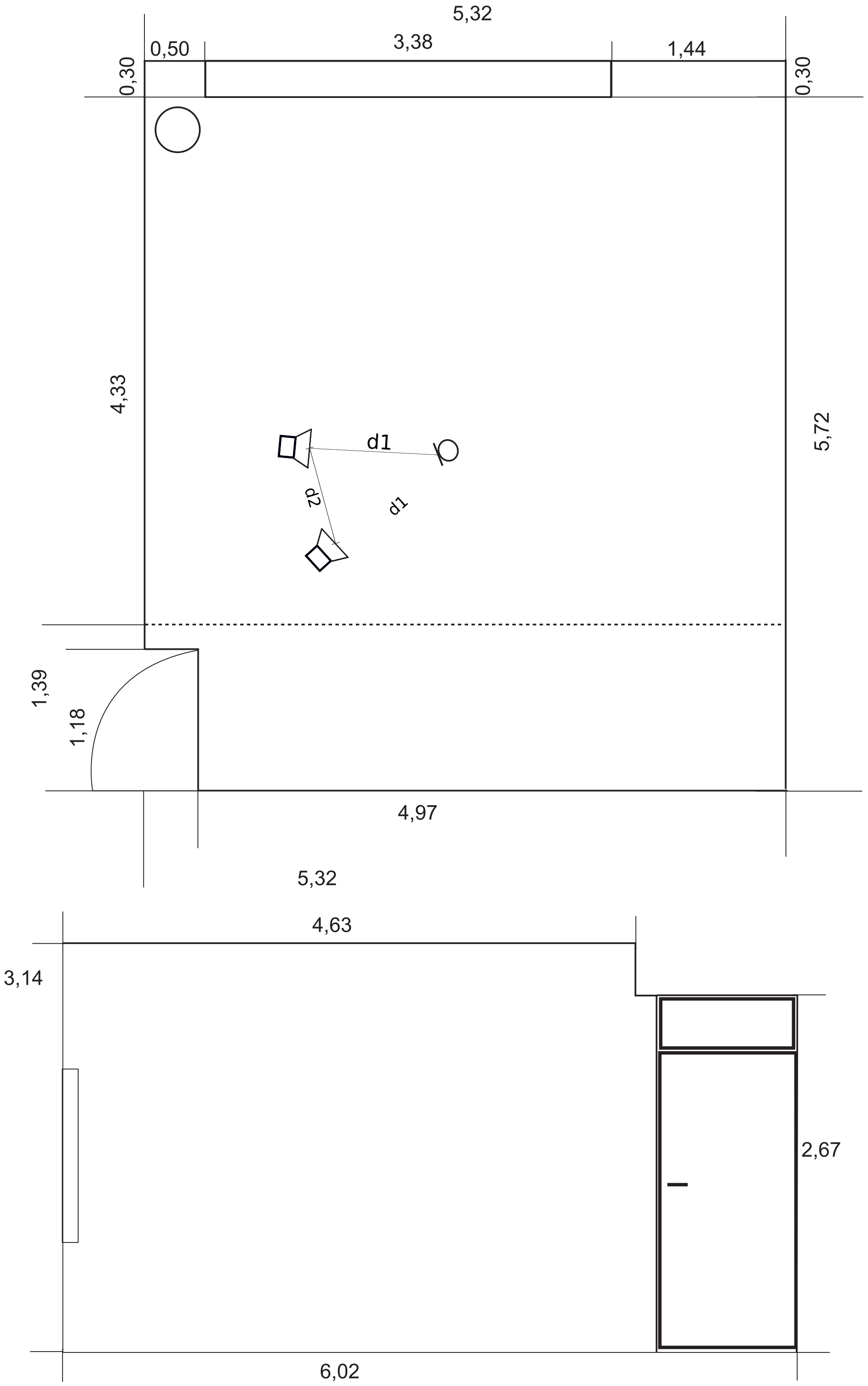


Figure A.1: Layout and dimensions of the Cocktail Party room. The ground plan is shown in the top plot and the vertical section in the bottom plot. The gap of 3.38 m indicates a window, the circle in the upper left corner represents a column (concrete), and between the column and the door their is a long bookshelf located. The whole floor is carpeted, otherwise the walls are empty.

well a line, since nobody was in the CPR room during the recording and the temperature also remained constant.

Table A.1: Recording setups for experiments in the cocktail party room. The distance $d1$ refers to the distance between loudspeakers and microphone, whereas the distance between loudspeakers is $d2$.

Recording setup	$d1$ [m]	$d2$ [m]
IR-25cm	0.25	0.45
IR-50cm	0.50	0.60
IR-100cm	1.00	0.80

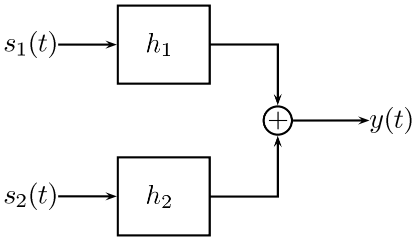


Figure A.2: Equivalent network of the speech mixture in a real environment. Symbol h_1 and h_2 model the respective transfer function from loudspeaker to microphone.

A.2 Recording Equipment

Two YAMAHA type MSP5 loudspeakers have been used for sound output. The speech material was collected from a single omni-directional Behringer ECM8000 measurement microphone. The captured signal was pre-amplified and digitized by the RME Fireface 800 at a sampling rate of 48 kHz. The *resample*[1] function of MATLAB has been used to upsample all database signals from 25 kHz to 48 kHz. Additionally, the signals were scaled to equal level. Furthermore, the output volume of the loudspeakers were adjusted to have the same level. The recording was carried out automatically, using the software Pure Data[2] (PD). Between each played back/recorded signal, a break of 1.5 sec. was specified manually. Thereby, it is ensured that the signal energy of the previous recording has decayed sufficiently. After recording, the signals were downsampled to a sampling frequency of 16 kHz by the *resample* function.

A.3 Results

For separation, the parallel gain-shape source-filter SCSS method proposed in chapter 6, section 6.3 has been utilized. Specifically, the parallel structure, where a Markov chain is associated to each hidden random variable is employed for separation. Moreover, the statistical parameters were optimized for clean speech and were not re-optimized for the recordings made in the CPR. This results in a significant mismatch between training and test data.

[1] http://www.mathworks.com/access/helpdesk/help/techdoc/ref/resampletimeseries.html

[2] http://puredata.info/

Figure A.3 shows the source separation results for the SGF, SGM, and DG case, separately. The labels on the horizontal-axis indicate different distances between loudspeakers and the microphone (see Table A.1). To assess performance, the separated signals are compared to the individually captured source signals, i.e. to the source signals convolved with the transfer function from loudspeaker to microphone. In our opinion, a comparison to the clean signals is not meaningful as the separation algorithm is not designed to compensate for introduced delay and spectral shaping by the room. Moreover, Rodrigues et al. [136] discussed the limitations of the binary mask for convolutive mixtures. According to their results, the binary mask improves the TMR and the intelligibility for rooms with reverberation time less than 300 ms. Note, the CPR has a reverberation time T_{60} [3] of $\sim$ 300 ms. Thus, according to these findings the binary mask is a less well suited operator for the separation of convolutive signal mixtures. Our results suggest that the application of the binary mask depends on the ratio of line-of-sight versus non-line-of-sight components for distances smaller than the critical distance [4]. For distances greater than the critical distance, we believe the T_{60} is an appropriate quantity to justify if the binary mask can improve the TMR and the intelligibility.

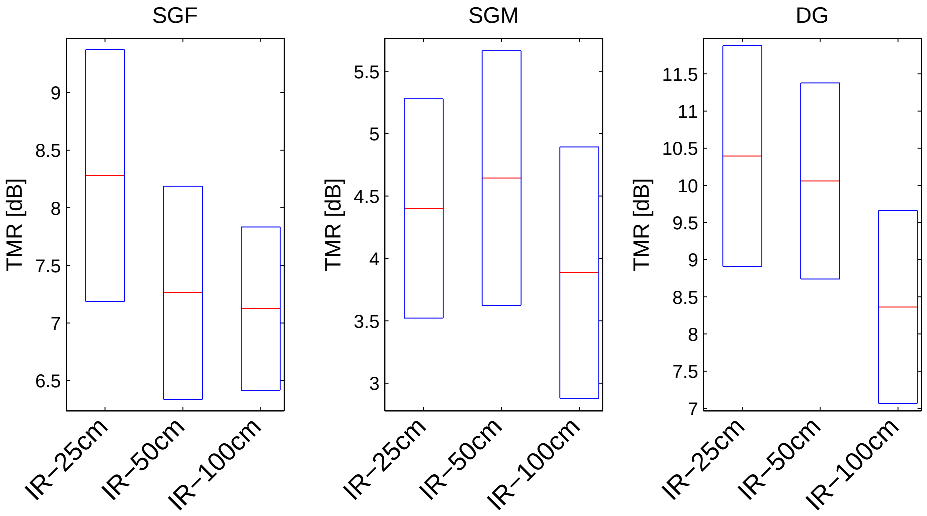


Figure A.3: Mean and standard deviation of the TMR for the BM signals using the parallel gain-shape source-filter based SCSS method. Specifically, the HMM-HMM structure of section 6.3 was used for separation. The labels indicate different distances between loudspeakers and microphone.

Comparing the results of “clean” co-channel speech (see Figure 6.22) to the data at distance of 25 cm, the TMR decreases by 2 dB and 1 dB for the SGF and SGM case, respectively. For the DG mixture no decrease in terms of TMR can be reported. Similarly, for the 50 cm distance experiment, the TMR for the DG case was slightly reduced and a little elevated for the SGM case. Only for the SGF case, TMR decreases noticeably to 7.3 dB. Finally, at a distance of 1 m, the TMR decreases for all cases. For this experiment, the impact of the room is already clearly audible.

[3] The reverberation time T_{60} is specified as the time duration by which the sound presure level of the impulse response drops by 60 dB.

[4] The distance at which the direct sound pressure level is equal to the reverberant sound pressure level is called the critical distance in acoustics.

A.4 Conclusion

In this section, the proposed parallel gain-shape source-filter single channel source separation method was applied to convolutive mixtures, recorded in a real environment. Therefore, the source signals have been played back by loudspeakers and afterwards captured by a microphone. Three experiments with different distances have been carried out to study the impact of the room on the separation performance. Currently, performance is measured comparing the separated signals to the source signals, also recorded in the room. For this setup we reported good results in terms of TMR. Especially, for the different gender case at distances of 25 and 50 cm no and only a minor decrease in TMR was achieved. Although this model does not include any convolutive approach even at a distance of 1 m, the TMR can still be improved by 7.1 dB, 3.8 dB, and 8.4 dB, for the SGF, SGM, and the DG case, respectively.

Appendix B

Automatic Speech Recognition

This chapter applies Single Channel Source Separation (SCSS) for automatic speech recognition applications. Specifically, a selection of discussed and developed algorithms are applied to speech mixtures of two databases. The first data set is the *SAIL real life SCSS corpus* [21] used in the next section. In section B.2, artificial mixtures are created from the Wall Street Journal corpora [38]. In both experiments we compare the three masks used for signal reconstruction of chapter 3, section 3.4.

B.1 ASR Results on the *SAIL Real Life Corpus*

In this experiment, we apply source separation to a real-world auditory scene instead of the artificial linear instantaneous mixture of speech signals. Therefore, we perform the separation experiments on the *SAIL real life SCSS corpus* [21], which is a compilation of different television newscasts. This corpus is split into a training and a test data set. The training set consists of clean speech material whereas the test set are mixture recordings containing spontaneous speech of two individuals and occasionally also laughter. Note, there is a considerable mismatch between the training data of predominantly read text and the test data. The test set consists of 13 utterances from four different individuals identified by the index: A-D. The utterances are identified by the speaker ID followed by an index number.

We compare the three masks used for signal reconstruction in chapter 3, section 3.4 in their application for speech recognition. During training, we learned a vector quantizer (VQ) with 265 codewords using approximately 2 min of speech material for each speaker. To separate the speech signals into its underlying source signals the factorial-max VQ model (FM-VQ) is used in combination with the ICM4 observation likelihood approximation of chapter 4, section 4.5. For the most probable state sequence, we reconstruct the separated signal using the binary mask (BM-VQ), the soft binary mask (SBM-VQ) and the soft mask (SM-VQ), respectively. Additionally, we present results for the non-negative matrix factorization method with sparseness constraints (NMF-l_0) [137]. We measure performance in terms of the achieved word error rate (WER) and compare the results of the three masks to each other and to the WER of the mixed signal. For performance assessment, we use the *Sail LABS Media Mining Indexer version 5.1* [22] ASR system. Table B.1 compares the resulting WER in [%] for the mixed signals, the NMF-l_0 method and the three different masks. The row at the bottom summarizes the average (Avg) WER over all utterances.

We see that only the BM-VQ is not able to decrease the average WER and that the SBM-VQ achieves the highest WER reduction. Moreover, non of the methods can decrease

Table B.1: Word error rate (WER) for the mixture and the separated signals using the NMF-l_0 and the factorial-max VQ method. For the factorial-max VQ three different mask for separation are employed: (i) binary mask (BM), (ii) soft binary mask (SBM), and (iii) soft mask (SM).

utterance	WER [%]				
	mixed	NMF-l_0	BM-VQ	SBM-VQ	SM-VQ
A1	87.5	87.5	87.5	87.5	87.5
A2	44.4	44.4	88.9	55.6	44.4
B1	33.3	44.4	55.6	0.0	0.0
B2	90.0	90.0	100.0	90.0	100.0
B3	93.3	80.0	100.0	93.3	100.0
B4	30.8	30.8	69.2	30.8	30.8
C1	80.0	80.0	100.0	80.0	80.0
C2	50.0	33.3	83.3	27.8	38.9
C3	55.0	70.0	100	55.0	60.0
C4	93.3	86.7	73.3	33.3	33.3
C5	85.7	92.9	85.7	78.6	78.6
D1	50.0	0.0	50.0	33.3	16.7
D2	60.0	20.0	40.0	0.0	0.0
Avg WER	**65.6**	**58.5**	**79.5**	**51.2**	**51.6**

the WER for the first speaker. These utterances contain purely spontaneous conversational speech with laughter, which is not included in the training material. If we compare the WER of the three different masks, the binary mask increases the WER in 8 utterances, the soft binary mask in one and the soft mask in three utterances. The WER remains unchanged for three, six and four utterances, respectively. We can report a decrease of WER for the binary mask, the soft binary mask and the soft mask for two, six and six utterances, respectively. These results suggest that the binary mask is not the best choice for speech separation in speech recognition applications. Moreover, the soft binary and the soft mask can improve the WER by almost the same amount. The SM-VQ however increases the WER for three utterances where the WER remains unchanged for the SBM-VQ. Both, the SBM-VQ and the SM-VQ can decrease the WER by a larger amount compared to the reference method NMF-l_0. Generally, these are just results of an small ASR experiment with only 13 utterances. Thus, these performance results have to be treated with great care.

B.2 ASR Results on the WSJ0 Nov92 Test Set

Due to the limited number of test data in the *SAIL Real Life SCSS Corpus* [21], we additionally perform ASR experiments on the November 1992 ARPA WSJ (Nov'92) test set of the Wall Street Journal corpora (WSJ0) [38]. We follow the HTK recipe of Vertanen [138] for training. Training and testing is performed using the HTK version 3.4. Before processing, the waveforms are parameterized into a feature vector of 39 dimensions consisting of 12 cepstral and the 0^{th} cepstral coefficients. Additionally, the delta and delta deltacoefficients are computed. The resulting Mel Frequency Cepstral Coefficients (MFCC) are normalized using cepstral mean subtraction. We use the TIMIT-bootstrapping method to initialize the 40 HMMs (39 HMMs for each phone and one HMM for silence) and follow the steps in the recipe to train the triphone models. For this training procedure only data of the training set in the WSJ0 corpora have been used.

The Nov'92 test set consists of 330 utterance from four female and four male speakers. The individuals are identified by their speaker ID: 440-447. This test set is evaluated using the WSJ 5K non-verbalized bigram language model.

Speech mixtures are created by artificially mixing each utterance of an individual with a randomly selected utterance of one of the other seven individuals. The length of the utterances is between two and ten seconds. This results in various overlap and non-overlap cases for the mixtures. For that reason, we adapt the calculation of the mixing level, i.e. the target-to-masker ratio (TMR), by normalizing the signal energies by their length:

$$\mathrm{TMR}_T = \frac{(\sum_{t=1} s_1(t)^2)/T_{s_1}}{(\sum_{t=1} s_2(t)^2)/T_{s_2}},$$

where $s_1(t)$ and $s_2(t)$ are the target and the interference, respectively. T_{s_1} and T_{s_2} are the corresponding utterance lengths in samples. In the experiment the most challenging case of equal mixing level has been selected, i.e. $\mathrm{TMR}_T = 0\ dB$. We present results in terms of the WER for every speaker separately, and a WER averaged (Avg WER) over all speakers. In the HTK book [139] the WER is defined as one minus the accuracy (ACC)

$$\mathrm{WER} = 1 - \mathrm{ACC} = \frac{D + S + I}{N} \times 100\%,$$

where N corresponds to the total number of labels in the reference transcription, D are the deletion errors, S are the substitution errors, and I are the insertion errors.

The first two columns of Table B.3 show the WER for the clean and the mixed database. The WER increases from 8.43 % for the clean to 106.93 % for the mixed database. Note, values greater than 100 % are possible due to the included insertion errors.

In the first experiment, we use the factorial-max VQ model (FM-VQ) in combination with the ICM4 observation likelihood approximation of chapter 4, section 4.5 to assess performance of the binary, soft binary, and soft mask. Therefore, we trained a speaker dependent model using each 5 min of data. Speaker dependent characteristics are represent using 500 codewords. The k-means training was stopped after 50 iterations. Column three to five of Table B.3 show the WER of the three masks. We notice that all three masks can decrease the WER compared to the WER of the mixture. As in the experiment of section B.2, the soft binary mask shows best performance with an average WER of 72.52 %. The soft mask performs slightly worse and the binary mask achieves only an average WER of 81.32 %. Note, the WER of the BM is approximately 10 % higher compared to the SBM mask although both masks use the same estimated state sequences. Note, the problem of optimal mask estimation for ASR has already been addressed in [68]. Additionally, we analyzed the considerable differences in WER between speakers. Especially the WER of the female speaker *445* is particularly low. The analysis of the speech mixture database showed no coherence between the ratio of same to different gender mixtures between target speakers and the WER. Table B.2 shows the percentage of same and different gender mixtures for each target speaker.

However, we found appreciable differences between target and interference utterance lengths. Only for the speaker *445* the average ratio of target to interference utterance length is as low as 0.72. This means that the target utterance lengths are on average only 72 % of the interference utterances. Moreover, the male speaker *440* has the highest utterance length ratio of 1.32, which explains for the lowest WER after source separation. For the remaining speakers the utterance length ratio is close to one.

Table B.2: Percentage of same gender (SGM) and different gender mixtures (DGM) for each target speaker in the database.

Speaker ID	440	441	442	443	444	445	446	447
SGM [%]	60.00	28.57	42.86	60.00	36.59	40.48	50.00	58.14
DGM [%]	40.00	71.43	57.14	40.00	63.41	59.52	50.00	41.86

Table B.3: Word error rate (WER) for the clean, mixed and the separated signals. For the factorial-max VQ three different masks for separation are employed: (i) binary mask (BM), (ii) soft binary mask (SBM), and (iii) soft mask (SM). For the source-filter (SF) and the gain-shape SF (GS-SF) method the SBM is used for separation.

Speaker ID	WER [%]						
	clean	mixed	BM-VQ	SBM-VQ	SM-VQ	SF	GS-SF
440	8.45	97.54	68.20	44.39	44.24	53.46	52.53
441	12.72	116.95	91.52	88.70	89.48	86.19	104.71
442	9.14	105.26	84.07	76.59	79.78	70.91	79.64
443	6.23	100.00	71.43	60.64	63.07	64.74	58.97
444	8.11	98.92	83.65	74.32	75.27	75.14	80.68
445	7.49	128.95	97.84	93.01	97.50	96.01	100.67
446	5.24	103.20	77.58	62.59	62.59	61.72	67.69
447	10.20	108.07	77.47	81.74	84.47	80.67	88.89
Avg WER	**8.43**	**106.93**	**81.32**	**72.52**	**74.29**	**73.27**	**78.89**

In the second experiment, we employ the parallel source-filter (SF) and the parallel gain-shape source-filter (GS-SF) separation methods of chapter 6, section 6.3 for ASR. We train one excitation and one filter related HMM for each speaker using 5 min of data. We use 500 states for each model. The HMMs of the GS-SF method are trained using data normalized to unit norm.

Since the three masks show the same behavior as for the FM-VQ method, we only present performance results using the SBM mask for reconstruction. ASR performance in terms of WER is shown in the last two columns of Table B.3. Although the SF method estimates two more parameters, it shows a negligible lower WER compared to the SBM-VQ method. Moreover, the GS-SF method, which additionally estimates the gain for each speaker at every time step, performs well with an WER of 78.89 %. Note, the WER for the GS-SF method is higher than for the other methods, but in contrast to these methods it can be applied to speech mixtures with arbitrary mixing level.

B.3 Conclusion

In this chapter we evaluated source separation methods for automatic speech recognition applications. We compared results in terms of the word error rate (WER) on two databases. The first is the *SAIL Real Life SCSS Corpus*, which is a compilation of television newscasts. This database consists of real world recordings of spontaneous speech. Moreover, we artificially mixed data of the November 1992 ARPA WSJ test set which is a part of the Wall Street Journal corpora. On both databases we conducted experiments using the factorial-max VQ. Specifically, we studied the impact of signal reconstruction on the WER using the binary, soft binary, and soft mask. On both databases, the soft binary mask showed superior performance while the binary mask seems less well suited for ASR applications. This is in contrast to Wang et al. [18, 19], who suggested the binary mask as computational goal in computational auditory scene analysis. Finally, we employed the proposed parallel source-filter (SF) and the parallel gain-shape source-filter (GS-SF) methods for separation. Although, the GS-SF method additionally estimates one more parameter, the performance is reasonable. In contrast to the factorial-max VQ and the SF method, the GS-SF method can be applied for arbitrary mixing levels. We conclude that all investigated separation methods can reduce the WER significantly compared to the WER without separation. However, all methods are far from the WER achieved on the clean database.

Bibliography

[1] F. Kschischang, B. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 498–519, 2001.

[2] D. Talkin, “Ch: A Robust Algorithm For Pitch Tracking,” In WB Kleijn and KK Paliwal editors, Speech Coding and Synthesis, pp. 495–518, 1995.

[3] D. Wang and G. J. Brown, Eds., *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications.* New Jersey: John Wiley and Sons Ltd, oct 2006.

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

[5] 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.

[6] D. S. Brungart, B. D. Simpson, M. A. Ericson, and K. R. Scott, “Informational and energetic masking effects in the perception of multiple simultaneous talkers,” *Journal of the Acoustical Society of America*, vol. 110, no. 5, pp. 2527–2538, Nov. 2001

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

[8] A. S. Bregman, *Auditory Scene Analysis: The Perceptual Organization of Sound*, 2nd ed. Cambridge: MIT Press, 1990.

[9] A. Hyvärinen and E. Oja, “Independent component analysis: Algorithms and applications,” *IEEE Int. Conf. Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[10] S. Choi, A. Cichocki, H.-M. Park, and S.-Y. Lee, “Blind source separation and independent component analysis: A review,” *Neural Information Processing - Letters and Reviews*, vol. 6, no. 1, pp. 1–57, January 2005.

[11] T. Quatieri and R. Danisewicz, “An approach to co-channel talker interference suppression using a sinusoidal model for speech,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 11-14 April 1988, pp. 565–568.

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

[13] G. Hu and D. Wang, “Monaural speech segregation based on pitch tracking and amplitude modulation,” *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1135–1150, 2004.

[14] Y. Shao and D. Wang, “Sequential organization of speech in computational auditory scene analysis,” *Speech Communication*, vol. 51, no. 8, pp. 657–667, Aug. 2009

[15] G. Hu and D. Wang, “Segregation of unvoiced speech from nonspeech interference,” *Journal of the Acoustical Society of America*, vol. 124, no. 2, pp. 1306–1319, 2008

[16] G. Hu, “Monaural speech organization and segregation,” Ph.D. dissertation, Ohio State University, 2006.

[17] S. T. Roweis, “One microphone source separation,” in *Neural Information Processing Systems, NIPS*, 2000, pp. 793–799

[18] G. Hu and D. Wang, “Speech segregation based on pitch tracking and amplitude modulation,” in *IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics*, New York, Oct. 2001, pp. 79–82.

[19] D. Wang, “Ch 12: On ideal binary mask as the computational goal of auditory scene analysis,” In P Divenyi editors, Speech Separation by Humans and Machines, pp. 181–197, 2004.

[20] M. P. Cooke, J. Barker, S. P. Cunningham, and X. Shao, “An audio-visual corpus for speech perception and automatic speech recognition,” in *Journal of the Acoustical Society of America*, ser. 5, vol. 120, 2006, pp. 2421–2424.

[21] J. Riedler and M. Stark, “A real life corpus for single channel source separation,” Graz University of Technology, SPSC Lab.; SAIL LABS Technology AG, Vienna, Tech. Rep. COAST-ROBUST-01, 2008.

[22] SAIL LABS Technology AG, “Media mining indexer (mmi),” Vienna, 2008.

[23] T. F. Quatieri, *Discrete-Time Speech Signal Processing*, ser. Prentice Hall series in signal processing. Upper Saddle River, NJ: Prentice Hall PTR, 2002.

[24] M. Radfar, R. Dansereau, and A. Sayadiyan, “Speaker-independent model-based single channel speech separation,” *Neurocomputing*, vol. 72, no. 1-3, pp. 71–78, Dec. 2008

[25] M. Radfar and R. Dansereau, “Single-channel speech separation using soft mask filtering,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2299–2310, 2007.

[26] M. Radfar, A. Sayadiyan, and R. Dansereau, “A new algorithm for two-speaker pitch tracking in single channel paradigm,” in *Proceedings of the 8th International Conference on Signal Processing*, vol. 1, China, Nov. 16-20 2006, pp. 1–4.

[27] M. Wu, D. Wang, and G. Brown, “A multipitch tracking algorithm for noisy speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 11, no. 3, pp. 229–241, 2003.

[28] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, “Monaural speech segregation based on fusion of source-driven with model-driven techniques,” *Speech Communication*, vol. 49, no. 6, pp. 464–476, Jun. 2007

[29] M. Wohlmayr and F. Pernkopf, “Multipitch tracking using a factorial hidden Markov models,” in *International Conference on Spoken Language Processing (Interspeech)*, 2008, pp. 147–150.

[30] ——, “Finite mixture spectrogram modeling for multipich tracking using a factorial hidden Markov model,” in *International Conference on Spoken Language Processing (Interspeech)*, 2009, pp. 1079–1082.

[31] M. Stark, M. Wohlmayer, and F. Pernkopf., “Source-filter based single channel speech separation using pitch information,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, Apr. 2010, to appear.

[32] M. Reyes-Gomez, “Statistical graphical models for scene analysis, source separation and other audio applications,” Ph.D. dissertation, Columbia University, New York, Dec. 2007, Department: Electrical Engineering

[33] M. Reyes-Gomez, D. Ellis, and N. Jojic, “Multiband audio modeling for single-channel acoustic source separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, Montreal, Canada, 2004, pp. 641–644.

[34] M. J. Reyes-Gomez, N. Jojic, and D. Ellis, “Deformable spectrograms,” *Artificial Intelligence and Statistics, AISTATS*, no. 79, pp. 1–26, Jan. 2005.

[35] M. Reyes-Gomez, N. Jojic, and D. P. W. Ellis, “Towards single-channel unsupervised source separation of speech mixtures: the layered harmonics/formants separation-tracking model,” in *Workshop on Statistical and Perceptual Audio Processing (SAPA)*, no. 137, Korea, 2004.

[36] S. J. Rennie, “Graphical models for robust speech recognition in adverse environments,” Ph.D. dissertation, Electrical and Computer Engineering, University of Toronto, 2008.

[37] M. Cooke, J. R. Hershey, and S. J. Rennie, “Monaural speech separation and recognition challenge,” *Computer Speech and Language*, vol. 24, no. 1, pp. 1–15, January 2010

[38] J. Garofalo, D. Graff, D. Paul, and D. Pallett, “Continous speech recognition (csr-i) wall street journal (wsj0) news, complete,” 1993, Linguistic Data Consortium, Philadelphia

[39] 3GPP, *Adaptive Multi-Rate Wideband Speech Transcoding, 3GPP TS 26.190*, 7th ed., ETSI, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex, Jun. 2007.

[40] D. O’Shaughnessy, “Invited paper: Automatic speech recognition: History, methods and challenges,” *Pattern Recognition*, vol. 41, no. 10, pp. 2965–2979, Oct. 2008

[41] R. Lippmann, “Speech recognition by machines and humans,” *Speech Communication*, vol. 22, no. 1, pp. 1–16, 1997

[42] A. W. Bronkhorst and R. Plomp, “Effect of multiple speechlike maskers on binaural speech recognition in normal and impaired hearing,” *Journal of the Acoustical Society of America*, vol. 92, no. 6, pp. 3132–3139, Dec. 1992

[43] V. Hamacher, J. Chalupper, J. Eggers, E. Fischer, U. Kornagel, H. Puder, and U. Rass, “Signal processing in high-end hearing aids: State of the art, challenges, and future trends,” *EURASIP Journal on Advances in Signal Processing*, vol. 2005, no. 18, pp. 2915–2929, 2005.

[44] S. Haykin and Z. Chen, “The cocktail party problem,” *Neural Computation*, vol. 17, no. 9, pp. 1875–1902, 2005

[45] F. J. Fraga, C. A. Ynoguti, and A. G. Chiovato, “Further investigations on the relationship between objective measures of speech quality and speech recognition rates in noisy environments,” in *International Conference on Spoken Language Processing (Interspeech)*, Pittsburgh, USA, September 2006, pp. 1877–1880.

[46] Y. Hu and P. Loizou, “Evaluation of objective quality measures for speech enhancement,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, 2008.

[47] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed. Prentice Hall, 1999.

[48] R. S. Bolia, W. T. Nelson, M. A. Ericson, and B. D. Simpson, “A speech corpus for multitalker communications research,” *Journal of the Acoustical Society of America*, vol. 107, no. 2, pp. 1065–1066, Feb. 2000

[49] M. Figueiredo and A. Jain, “Unsupervised learning of finite mixture models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, 2002.

[50] M. Sabin and R. Gray, “Product code vector quantizers for waveform and voice coding,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 3, pp. 474–488, 1984.

[51] F. Jelinek, *Statistical Methods for Speech Recognition.* MIT Press, January 1998.

[52] E. Ristad, “A natural law of succession,” Dept. of Computer Science, Princeton University, Princeton, NJ, Research Report CS-TR-495-95, 1995.

[53] C. Von der Malsburg, “The correlation theory of brain function,” in *Reprinted in: Models of Neural Networks II*, E. Domany, J. V. Hemmen, and K. Schulten, Eds. Berlin: Springer, 1981, ch. 2, pp. 95–119

[54] A. J. W. van der Kouwe, D. Wang, and G. J. Brown, “A comparison of auditory and blind separation techniques for speech segregation,” *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 189–195, Mar. 2001.

[55] B. Sagi, S. C. Nemat-Nasser, R. Kerr, R. Hayek, C. Downing, and R. Hecht-Nielsen, “A biologically motivated solution to the cocktail party problem,” *Neural Computation*, vol. 13, no. 7, pp. 1575–1602, 2001

[56] H. Asari, B. A. Pearlmutter, and A. M. Zador, “Sparse representations for the cocktail party problem,” *Journal of Neuroscience*, vol. 26, no. 28, pp. 7477–7490, 2006

[57] R. Pichevar, J. Rouat, C. Feldbauer, and G. Kubin, “A bio-inspired sound source separation technique in combination with an enhanced FIR gammatone analysis/synthesis filterbank,” in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Vienna, Sep. 2004, pp. 2063–2066.

[58] I. Jolliffe, *Principal Component Analysis.* New York: Springer, 1986.

[59] G.-J. Jang, T.-W. Lee, and Y.-H. Oh, “Blind separation of single channel mixture using ICA basis functions,” in 3^{rd} *International Conference on ICA and BSS (ICA2001)*, San Diego, CA, USA, Dec. 2001, pp. 9–12.

[60] A. Hyvärinen, “Fast and robust fixed-point algorithms for independent component analysis,” *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[61] A. Hyvärinen and P. Hoyer, “Emergence of phase- and shift-invariant features by decomposition of natural images into independent feature subspaces,” *Neural Computation*, vol. 12, no. 7, pp. 1705–1720, Jul. 2000

[62] M. A. Casey, “Separation of mixed audio sources by independent subspace analysis,” Merl - A Mitsubishi Electric Research Laboratory, Massachusetts, USA, Tech. Rep. TR-2001-31, Sep. 2001.

[63] G.-J. Jang and T.-W. Lee, “A maximum likelihood approach to single-channel source separation,” *Journal of Machine Learning Research*, vol. 4, no. 7-8, pp. 1365–1392, 2003

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

[65] D. Lee and H. Seung, “Learning the parts of objects by nonnegative matrix factorization,” *Nature*, vol. 401, pp. 788–791, August 1999.

[66] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Advances in Neural Information Processing Systems (NIPS).* MIT Press, 2000, pp. 556–562.

[67] P. Smaragdis and J. Brown, “Non-negative matrix factorization for polyphonic music transcription,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, Oct. 2003, pp. 177–180.

[68] B. Raj, R. Singh, and P. Smaragdis, “Recognizing speech from simultaneous speakers,” in *International Conference on Spoken Language Processing (Interspeech)*, Lisbon, September 2005, pp. 3317–3320.

[69] S. T. Roweis, “Factorial models and refiltering for speech separation and denoising,” in *Proceedings of the European Conference on Speech Communication and Technology*, Geneva, Switzerland, Sep. 2003, pp. 1009–1012.

[70] P. Smaragdis, *Non-negative Matrix Factor Deconvolution; Extraction of Multiple Sound Sources from Monophonic Inputs.* Springer Berlin/Heidelberg, 2004

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

[72] B. Raj and P. Smaragdis, "Latent variable decomposition of spectrograms for single channel speaker separation," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 17–20.

[73] B. Raj, P. Smaragdis, M. Shashanka, and R. Singh, "Separating a foreground singer from background music," in *Proceedings, International Symposium on Frontiers of Research on Speech and Music (FRSM)*, Mysore, India, Jan. 2007.

[74] T. Dau, D. Puschel, and A. Kohlrausch, "A quantitative model of the "effective" signal processing in the auditory system.: I. Model structure," *Journal of the Acoustical Society of America*, vol. 99, no. 6, pp. 3615–3622, Jun. 1996

[75] S. Schimmel, L. Atlas, and K. Nie, "Feasibility of single channel speaker separation based on modulation frequency analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2007, pp. 605–608.

[76] M. H. Radfar, R. M. Dansereau, and A. Sayadiyan, "A maximum likelihood estimation of vocal-tract-related filter characteristics for single channel speech separation," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 1, pp. 1–15, 2007.

[77] J. H. L. Hansen and B. L. Pellom, "An effective quality evaluation protocol for speech enhancement algorithms," in *Proceedings of the International Conference on Spoken Language Processing*, no. 0917, 1998, pp. 2819–2822.

[78] P. Mermelstein, "Evaluation of a segmental SNR measure as an indicator of the quality of ADPCM coded speech," *Journal of the Acoustical Society of America*, vol. 66, no. 6, pp. 1664–1667, Dec. 1979

[79] R. Gribonval, L. Benaroya, E. Vincent, and C. Fvotte, "Proposals for performance measurement in source separation," in 4^{th} *Int. Symp. on Independent Component Analysis and Blind Source Separation (ICA)*, Nara, Japan, Apr. 2003, pp. 715–720.

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

[81] R. Lambert and A. Bell, "Blind separation of multiple speakers in a multipath environment," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, Munich, Germany, 1997, pp. 423–426.

[82] D. Schobben, K. Torkkola, and P. Smaragdis, "Evaluation of blind signal separation methods," in *Int. Workshop on ICA and Blind Signal Separation*, Aussois, France, January 11-15, 1999, pp. 261–266.

[83] T. Takatani, T. Nishikawa, H. Saruwatari, and K. Shikano, "Simo-model-based independent component analysis for high-fidelity blind separation of acoustic signals," in 4^{th} *Int. Symp. on Independent Component Analysis and Blind Signal Separation*, Nara, Japan, April 2003, pp. 993–998.

[84] L. Benaroya, F. Bimbot, and R. Gribonval, "Audio source separation with a single sensor," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 191–199, 2006.

[85] D. P. Ellis, “Ch 20: Evaluating Speech Separation Systems,” In P Divenyi editors, Speech Separation by Humans and Machines, pp. 295–304, 2004.

[86] T. O. Virtanen, “Speech recognition using factorial Hidden Markov Models for separation in the feature space,” in *International Conference on Spoken Language Processing (Interspeech).* Pittsburgh: ISCA, September 2006, pp. 89–92.

[87] A. Deoras and A. Hasegawa-Johnson, “A factorial HMM approach to simultaneous recognition of isolated digits spoken by multiple talkers on one audio channels,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, May 2004, pp. 861–864.

[88] D. P. Ellis, “Prediction-driven computational auditory scene analysis,” Ph.D. dissertation, Massachusetts Institute of Technology, June 1996.

[89] R. C. Bilger, J. M. Nuetzel, W. M. Rabinowitz, and C. Rzeczkowski, “Standardization of a test of speech perception in noise,” *Journal of Speech and Hearing Research*, vol. 27, no. 1, pp. 32–48, Mar. 1984

[90] M. Nilsson, S. Soli, and J. Sullivan, “Developement of the hearing in noise test for the measurement of speech reception threshold in quiet and in noise,” in *Journal of the Acoustical Society of America*, vol. 95, 1994, pp. 1085–1099.

[91] *ITU-T Recomendation P.800: Methods for Subjective Determination of Transmission Quality*, International Telecommunication Union-Telecom, Geneva, March 1996.

[92] P. C. Loizou, *Speech Enhancement, Theory and Praxis.* Boca Raton: CRC Press, Taylor & Francis Group, 2007.

[93] Z. Ghahramani and M. Jordan, “Factorial hidden Markov models,” *Machine Learning*, vol. 29, no. 2-3, pp. 245–273, 1997.

[94] M. Jordan, *Learning in Graphical Models.* MIT Press, 1999.

[95] A. Nadas, D. Nahamoo, and M. A. Picheny, “Speech recognition using noise-adaptive prototypes,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 37, no. 10, pp. 1495–1503, Oct. 1989.

[96] A. Varga and R. Moore, “Hidden Markov Model decomposition of speech and noise,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, 1990, pp. 845–848.

[97] B. J. Frey, L. Deng, A. Acero, and T. Kristjansson, “Algonquin: Iterating Laplace’s method to remove multiple types of acoustic distortion for robust speech recognition,” in *Proceedings of the European Conference on Speech Communication and Technology*, 2001, pp. 901–904.

[98] H. Pobloth and W. B. Kleijn, “Squared error as a measure of perceived phase distortion,” *Journal of the Acoustical Society of America*, vol. 114, no. 2, pp. 1081–1094, Aug. 2003

[99] M. Radfar, A. Banihashemi, R. Dansereau, and A. Sayadiyan, “Nonlinear minimum mean square error estimator for mixture-maximisation approximation,” *Electronics Letters*, vol. 42, no. 12, pp. 724–725, 2006.

[100] M. Wohlmayr, M. Stark, and F. Pernkopf, “A probabilistic interaction model for multipitch tracking with factorial Hidden Markov Models,” *IEEE Transactions on Audio, Speech, and Language Processing*, 2010, sumitted.

[101] T. Kristjansson, H. Attias, and J. Hershey, “Single microphone source separation using high resolution signal reconstruction,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, 2004, pp. 817–20.

[102] D. Klatt, “A digital filter bank for spectral matching,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, 1976, pp. 573–576.

[103] A. M. Reddy and B. Raj, “A minimum mean squared error estimator for single channel speaker separation,” in *International Conference on Spoken Language Processing (Interspeech - ICSLP)*, 2004, pp. 2445–2448.

[104] ——, “Soft mask methods for single-channel speaker separation,” in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 6, Aug. 2007, pp. 1766–1776.

[105] S. J. Rennie, J. R. Hershey, and P. A. Olsen, “Single-channel speech separation and recognition using loopy belief propagation,” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 3845–3848, 2009.

[106] C. M. Bishop, *Pattern Recognition and Machine Learning.* 233 Spring Street, New York, NY 10013, USA: Springer, 2006.

[107] M. Stark and F. Pernkopf, “On optimizing the computational complexity for VQ-based single channel source separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, Texas, 2010, pp. 237–240.

[108] E. Bocchieri, “Vector quantization for the efficient computation of continuous density likelihoods,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, Minneapolis, USA, Apr. 1993, pp. 692–695.

[109] A. Viterbi, “Error bounds for convolutional codes and an asymptotically optimum decoding algorithm,” *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[110] M. Stark and F. Pernkopf, “Towards source-filter based single sensor speech separation,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Taipei, Taiwan, April 2009, pp. 97 – 100.

[111] A. Papoulis and U. Pillai, S., *Probability, Random Variables and Stochastic Processes*, 4th ed. McGraw-Hill, 2002, ch. 15: Markov Chains, pp. 695–772.

[112] J. Besag, “On the statistical analysis of dirty pixtures,” *Journal of the Royal Statistical Society*, vol. 48, no. 3, pp. 259–302, 1986.

[113] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[114] C. Lawson and R. Hanson, *Solving Least-Squares Problems.* Prentice-Hall, 1974, ch. Linear Least Squares with linear inequality constraints, pp. 158–166.

[115] S. M. Kay, *Fundamentals of Statistical Signal Processing, Estimation Theory*, ser. Prentice Hall signal processing series. PTR Prentice-Hall, 1993, vol. 1.

[116] J. Kivinen and M. K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," in *Proceedings of the* 27^{th} *Annual ACM Symposium on Theory of Computing*. Las Vegas, Nevada, United States: ACM, 1995, pp. 209–218.

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

[118] J. B. Buckheit and D. L. Donoho, "WaveLab and reproducible research," Stanford University, Stanford, CA, USA, Tech. Rep., 1995.

[119] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, ser. The Springer International Series in Engineering and Computer Science. Springer, 1992, vol. 159, ch. Constraint Vector Quantization, pp. 407–482.

[120] M. Stark, F. Pernkopf, T. V. Pham, and G. Kubin, "Vocal-tract modeling for speaker independent single channel source separation," in 1^{st} *IAPR Workshop on Cognitive Information Processing*, Santorini, Greece, June 2008, pp. 217 – 220.

[121] P. Vary and R. Martin, *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. John Wiley, march 2006.

[122] D. Morgan, E. George, L. Lee, and S. Kay, "Cochannel speaker separation by harmonic enhancement and suppression," *IEEE Transactions on Speech and Audio Processing*, vol. 5, no. 5, pp. 407–424, 1997.

[123] J. Rosier and Y. Grenier, "Two-pitch estimation for co-channel speakers separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 13-17 May 2002, pp. 4–4160.

[124] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *Journal of the Royal Statistic Society*, vol. 30, no. B, pp. 1–38, 1977.

[125] M. H. Hansen and B. Yu, "Model selection and the principle of minimum description length," *Journal of the Acoustical Society of America*, vol. 96, no. 454, pp. 746–774, Jun., 2001

[126] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1344–1348, 2005.

[127] T. Minka, "Divergence measures and message passing," Microsoft Research Cambridge, Tech. Rep. MSR-TR-2005-173, 2005.

[128] J. Laroche, Y. Stylianou, and E. Moulines, "HNS: Speech modification based on a harmonic+noise models," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2, 27-30 April 1993, pp. 550–553.

[129] R. McAulay and T. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 34, no. 4, pp. 744–754, 1986.

[130] D. Paul, “The spectral envelope estimation vocoder,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 4, pp. 786–794, 1981.

[131] R. McAulay and T. Quatieri, “Ch: Sinusoidal Coding,” In WB Kleijn and KK Paliwal editors, Speech Coding and Synthesis, New York, USA, pp. 121–173, nov, 1995.

[132] J. Makhoul, “Linear prediction: A tutorial review,” *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, April 1975.

[133] J. Bilmes and G. Zweig, “The graphical models toolkit: An open source software system for speech and time-series processing,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 4, 2002, pp. 3916–3919.

[134] A. Wrench, “A multichannel/multispeaker articulatory database for continuous speech recognition research,” in *Workshop on Phonetics and Phonology in ASR*, vol. 5, 2000, pp. 3–17

[135] L. Parra and C. Spence, “Convolutive blind separation of non-stationary sources,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 3, pp. 320–327, 2000.

[136] G. Rodrigues and H. Yehia, “Limitations of the spectrum masking technique for blind source separation,” *Independent Component Analysis and Signal Separation*, pp. 621–628, 2009

[137] R. Peharz, “Single channel source separation using dictionary design methods for sparse coders,” Master’s thesis, Graz Technical Unicersity of Technology, march 2010.

[138] K. Vertanen, “Baseline WSJ acoustic models for HTK and Sphinx: Training recipes and recognition experiments,” Cavendish Laboratory, University of Cambridge, Tech. Rep., 2006.

[139] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for HTK Version 3.4)*. Microsoft Corporation/Cambridge University Engineering Department, 2006.