KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Joint speaker localization, enhancement and identification: A factorization-based approach

Bart Gesquiere

Thesis submitted for the degree of
Master of Science in
Electrical Engineering, option
Embedded Systems and Multimedia

Thesis supervisor:
Prof. dr. ir. Hugo Van hamme

Academic year 2014 – 2015

Master of Science in Electrical Engineering

KU LEUVEN

FACULTEIT INGENIEURSWETENSCHAPPEN

Joint speaker localization, enhancement and identification: A factorization-based approach

Bart Gesquiere

Thesis submitted for the degree of
Master of Science in
Electrical Engineering, option
Embedded Systems and Multimedia

Thesis supervisor:
Prof. dr. ir. Hugo Van hamme

Assessors:
Prof. dr. ir. Marc Moonen
Prof. dr. ir. Dirk Van Compernolle

Mentor:
dr. ir. Sayeh Mirzaei

Academic year 2014 – 2015

Voorwoord

Deze thesis is het resultaat van lange dagen met veel *gesakker*, maar nu kan ik trots terugkijken op het voorbije jaar. Zonder een paar mensen zou deze thesis niet zijn wat ze nu is. Aan deze mensen wil ik hier mijn oprechte dank betuigen.

Graag wil ik Malcolm bedanken voor het nalezen van de thesis. Graag zou ik mijn begeleidster, Sayeh, willen bedanken. Zelfs al had ze het druk met haar doctoraatsthesis en was ze het voorbije jaar in het buitenland, nog steeds vond ze de tijd om me te helpen: *thank you, Sayeh!* Graag zou ik ook Jay willen bedanken die me toestemming heeft gegeven om de door hem opgenomen *Panel Meeting* data set te gebruiken: *thank you, Jay!* Ik zou vooral ook mijn dank willen betuigen aan mijn promotor, Prof. Hugo Van hamme. Niet alleen heeft hij het onderwerp voor deze thesis aangereikt, ik kon ook steeds bij hem terecht voor een verhelderend gesprek in een zeer aangename sfeer. Dank u wel daarvoor.

Deze thesis vormt het sluitstuk van mijn studententijd. Ik zou daarom graag de tijd willen nemen om ook een paar andere mensen te bedanken. Graag zou ik mijn vrienden willen bedanken, voor de vele mooie momenten van de voorbije jaren die ik nooit zal vergeten. Ik wil mijn vader bedanken, omdat hij zich zijn hele leven kapot heeft gewerkt om me de kans te geven van mijn toekomst te maken wat ik er zelf van wil maken. Mijn moeder wil ik bedanken, omdat ze me er steeds toe heeft aangezet het beste van mezelf te geven. Ik zou ook graag mijn zus willen bedanken, omdat ik al heel mijn leven lang op haar kan rekenen.

Maar bovenal wil ik mijn vriendin, Orpha, bedanken. Het voorbije jaar stond ze steeds klaar aan mijn zijde om mijn twijfels de kop in te drukken en me de moed en energie te geven om er volledig voor te blijven gaan. Zonder jou was ik waarschijnlijk een beetje gek geworden. Bedankt voor alles, liefste.

Bart Gesquiere

Abstract

In this thesis, we propose a new method for object-based speaker identification based on nonnegative matrix factorization. We assume that a magnitude spectrogram $\mathbf{V_s}$ from a single-speaker speech segment can be characterized by a limited set of objects. The goal of nonnegative matrix factorization is to factorize each speaker spectrogram $\mathbf{V_s}$ into a dictionary $\mathbf{W_s}$, which contains a set of characteristic objects, and an activation matrix $\mathbf{H_s}$, which contains weights for each object per frame. In this work, we apply a variant of nonnegative matrix factorization incorporating Bayesian inference in which we assume that objects are generated according to a gamma-Poisson generative model. Additionally, a technique for model order estimation is adopted in which relevance parameters denote which objects are relevant and which objects should be pruned away. As a result, speaker dictionaries $\mathbf{W_s}$ are obtained which are characteristic to speakers without risk of overfitting $\mathbf{V_s}$.

Once different speaker dictionaries $\mathbf{W_s}$ have been extracted, we propose to extract features based on averaged activations $\mathbf{h_n}$ obtained from group sparsity nonnegative matrix factorization with a combined dictionary $\mathbf{W_{TOT}}$. This combined dictionary contains every object from all known speakers. Group sparsity nonnegative matrix factorization enables the definition of groups of objects within this combined dictionary corresponding to speakers and finds solutions where only a limited set of groups is active. We show that when such activations are averaged over a window of 0.5 seconds, features are obtained which achieve competitive results compared to other object-based speaker identification for high quality recordings. We also show that our technique is applicable to noisy field recordings of panel meetings, albeit at degraded performance. The key advantage of our object-based speaker identification with respect to traditional methods is that the development of a universal background model is not needed.

However, due to our chosen classification procedure, our object-based speaker identification method is only applicable to single-speaker segments. We alleviate this problem by adopting techniques for blind source separation and localization and using a microphone array with two microphones. Speech enhancement is achieved through factorization-based source separation of possibly underdetermined signals using an EM algorithm as proposed by Ozerov *et al.* (2010,[1]). However, initial estimates of the parameters are needed in order to execute the EM algorithm. Mirzaei *et al.* have proposed an initialization scheme where an initial estimate for the mixing matrix is obtained by counting and locating speakers and initial estimates for the speaker spectrograms are obtained with binary masking (2014, [2]). In this work, we show that our joint system shows excellent performance for non-reverberated signals and when at most two speakers speak simultaneously. However, when more than two speakers speak concurrently or when recordings contain reverberation, the performance of the joint system degrades.

Samenvatting

In deze thesis stellen we een op object gebaseerde methode voor sprekerherkenning voor, die gebaseerd is op niet-negatieve matrix factorisatie. We veronderstellen dat een magnitude spectrogram $\mathbf{V_s}$ van een opname van een spreker gekenmerkt kan worden door een beperkt aantal objecten. Het doel van niet-negatieve matrix factorisatie is om elk spectrogram $\mathbf{V_s}$ te ontbinden in een woordenboek $\mathbf{W_s}$, dat de objecten bevat die de spreker kenmerkt, en een activatiematrix $\mathbf{H_s}$, die de gewichten bevat voor elk object per venster. In deze thesis passen we een Bayesiaanse variant van niet-negatieve matrix factorisatie toe waarin we veronderstellen dat objecten gegenereerd worden volgens een gamma-Poisson statistisch model. Bovendien passen we een techniek toe voor het schatten van de orde van de ontbinding door relevantieparameters te introduceren die aangeven welke objecten relevant zijn en welke niet.

Indien de woordenboeken beschikbaar zijn, kunnen *features* geëxtraheerd worden via niet-negatieve matrix factorisatie met *group sparsity*. Deze techniek gebruikt het gecombineerde woordenboek $\mathbf{W_{TOT}}$ dat elk object bevat van alle gekende sprekers. Groepen die overeenkomen met sprekers kunnen gedefinieerd worden binnen dit gecombineerde woordenboek. Via niet-negatieve matrix factorisatie met *group sparsity* wordt een activatiematrix gevonden, waarin slechts een beperkte set groepen actief is. Wanneer deze methode wordt toegepast op segmenten van 0.5 seconden en de resulterende activaties uitgemiddeld worden over de vensters van dit segment, verkrijgen we *features* die competitieve resultaten opleveren. We tonen ook aan dat onze methode toepasbaar is op opnames van vergaderingen met ruis, hoewel dit een slechtere performantie oplevert. Het grootste voordeel van onze op object gebaseerde methode voor sprekerherkenning ten opzichte van traditionele methoden, is dat een *universal background model* niet nodig is.

Door de gekozen classificatie is onze methode slechts toepasbaar op segmenten met slechts één spreker. Dit probleem wordt opgelost door bestaande technieken voor localisatie en bronscheiding toe te passen en gebruik te maken van een microfoonarray met twee microfoons. Bronscheiding gebeurt door een op ontbinding gebaseerd EM algoritme dat ontwikkeld is door Ozerov *et al.* (2010, [1]). Deze techniek is eveneens toepasbaar indien het aantal aanwezige bronnen hoger is dan het aantal beschikbare microfoons. Als initialisatie voor de parameters van de bronscheiding wordt een door Mirzaei *et al.* voorgestelde initialisatieprocedure toegepast. Hierin wordt de *mixing matrix* geschat door sprekers te tellen en te localiseren en worden de bronspectrogrammen $\mathbf{V_s}$ geschat door middel van *binary masking* (2014, [2]). In deze thesis tonen we aan dat het gezamenlijke systeem uitstekend presteert indien er geen galm aanwezig is en indien er hoogstens twee sprekers gelijktijdig spreken. Indien er meer dan twee sprekers gelijktijdig spreken en vooral indien er galm aanwezig is daalt de performantie van het gezamenlijke systeem.

Contents

Nomenclature

List of Abbreviations

BA	Bayesian Adaptation
BNMF	Bayesian Nonnegative Matrix Factorization
DFT	Discrete Fourier transform
EM	Expectation-Maximization
EUC	Euclidean distance measure
FFT	Fast Fourier Transform
GaP	Gamma-Poisson
GCC-PHAT	Generalized Cross Correlation with Phase Transform
GS-NMF	Group Sparsity Nonnegative Matrix Factorization
GMM-UBM	Gaussian Mixture Model with Universal Background Model
IS	Itakura-Saito divergence
KL	Kullback-Leibler divergence
MFC	Mel Frequency Coefficients
MFCC	Mel Frequency Cepstrals Coefficients
NMF	Nonnegative Matrix Factorization
PCA	Principal Component Analysis
RIR	Room Impulse Response
STFT	Short Time Frequency Transform
SAR	Signal-To-Artifact Ratio
SDR	Signal-To-Distortion Ratio
SIR	Signal-To-Inference Ratio
SVM	Support Vector Machine
SVM-GSV	Support Vector Machine for GMM SuperVectors
UBM	Universal Background Model
VBEM	Variational Bayesian Expectation-Maximization
VQ	Vector Quantization

List of Symbols

$\mathbf{a_{speq}}$	angular spectrum
A	# of angles
$\mathbf{A}$	mixing matrix
$\mathbf{\mathring{A}_f}$	the augmented mixing matrix
$\mathbf{A_{EST}}$	matrix with spatiotemporal detections
$\mathbf{b_{fn}}$	channel noise of all channels for a time-frequency bin
$\mathbf{C}$	set of all components
$\mathbf{C_k}$	component
$f_{c,m}$	central frequency of the m^{th} mel filter
$f_{freq,f}$	frequency corresponding to frequency bin f
F	# of frequency bins
$g_{v_1v_2}(f)$	cross power spectral density
$\mathbf{G_B}$	group allocation matrix
$\mathbf{H}$	(time) activation matrix
$\mathbf{H_{GS}}$	(time) activation matrix with group sparsity
I	# of channels
J	# of frames per block (GS-NMF)
$k(x_1, x_2)$	kernel
K	model order
K^*	inherent model order for $\mathbf{V}$
$\mathbf{L}$	localization matrix
$\mathbf{L}'$	smoothed localization matrix
M	# of mel bins or # Gaussians per mixture model
N	# of frames or observations in $\mathbf{V}$
N_{FFT}	# of samples per time-domain frame
N_{iter}	# of iterations
P	# of simultaneous segments
$q(\cdot)$	auxiliary/free distribution
$r^{\boldsymbol{\mu}}$	relevance parameter (GMM-UBM)
$r_{v_1v_2}(\tau)$	cross-correlation
$\mathbf{T}$	frequency-to-mel transformation matrix or total variability matrix (i-vectors)
$\mathbf{V}$	data (a spectrogram) to be factorized
$\mathbf{V_{freq}}$	magnitude (frequency) spectrogram
$\mathbf{V_{mel}}$	mel spectrogram
$\mathbf{V_{VAD}}$	spectrogram after voice activity detection
$\mathbf{V_i^c}$	channel spectrogram as recorded by the i^{th} microphone
$\mathbf{V_s}$	source spectrogram from speaker s
$\mathbf{V^{BM}}$	initial estimate for a source spectrogram (binary masking)
$\mathbf{V^{SS}}$	separated spectrogram (BSS)
w_i	mixture weight for mixture i
$\mathbf{W}$	dictionary with objects w_k in its columns
$\mathbf{W_{TOT}}$	combined dictionary
$\mathbf{x}$	feature vector
$\mathbf{X}$	sequence of feature vectors
$\mathbf{y}$	label

α	GCC-PHAT parameter for non-linear transformation of the angular spectrum
$\alpha_i^{\boldsymbol{\mu}}$	data-dependent adaptation coefficient for mixture i
$\Delta_{A,min}$	minimum angular separation
$\Delta_{N,min}$	minimum segmental length in frames
$\Delta_{N,max}$	maximum intersegmental silence
$\Gamma(x)$	gamma function
$\boldsymbol{\lambda}$	parameters of a Gaussian mixture model or vector of relevance parameters
λ_1	sparsity parameter
λ_g	group sparsity parameter
Λ	log-likelihood ratio (GMM-UBM)
Λ_{WH}	log-likelihood of $\mathbf{W}$ and $\mathbf{H}$
$\Lambda_{\text{WH,LB}}$	lower bound to the log-likelihood of $\mathbf{W}$ and $\mathbf{H}$
$\boldsymbol{\mu_i}$	mean for mixture i
$\boldsymbol{\Sigma_i}$	covariance matrix for mixture i
τ	threshold or time delay
$\phi(x)$	non-linear transformation (SVM)
φ	automatic relevance determination parameter
$\mathscr{G}(x\|a,b)$	gamma distribution with hyperparameters a and b
$\mathcal{N}(x\|a,b)$	Gaussian distribution with mean a and variance b
$\mathscr{P}(x\|\lambda)$	Poisson distribution with intensity parameter λ

List of Figures

List of Tables

Chapter 1

Introduction

Speech is a natural form of communication, which is why there is such interest in machine learning techniques for automation tasks related to speech. Aided by the rapid developments in the digital world in the past decades, the way has been paved for a lot of novel speech processing techniques. Some of these include but are not limited to speech detection—the task of determining whether someone is speaking; gender identification—the task of labelling an utterance with the correct gender; language recognition—determining the language spoken in a speech segment; speech recognition—the task of determining which words are spoken; and speaker recognition—the task of extracting information about the identity of a speaker when provided with a speech segment.

In this thesis the focus will lie on **speaker recognition** which has important applications in the fields of e.g. authentication, surveillance, security, forensics and multi-speaker tracking [3]. Our goal in this thesis is to apply an existing technique, nonnegative matrix factorization, in the context of speaker recognition. Nonnegative matrix factorization has successfully been applied to automatic music transcription [4] and speech recognition [5, 6]. However, it has only recently been applied to speaker recognition. To the best of our knowledge, it has only been explored in one research paper by Joder *et al.* [7][1]. Our goal is to define novel object-based features for speaker recognition method that differ from those defined by Joder *et al.* [7]. Our hope is that this thesis can contribute to the Cametron project. The goal of this project is to build a system which uses state-of-the-art audiovisual technologies for producing high quality audiovisual productions [8]. Our method for object-based speaker identification can be combined with blind source separation as proposed by Ozerov *et al.* with an improved initialization scheme as proposed by Mirzaei *et al.* [1, 2]. This joint system can be applied within the Cametron project for joint speaker localization, enhancement and identification. The remainder of this thesis is organized as follows.

In chapter 2, an overview of the state-of-the-art in the field of speaker recognition is given. This chapter includes concise explanations for existing speaker recognition methods such as GMM-UBM, SVM-GSV and i-vectors. These methods serve as a reference to compare our method for speaker identification against in the following chapters.

In chapter 3, an overview is given of the main techniques that are necessary for our speaker identification method such as nonnegative matrix factorization and a variant with Bayesian inference. The technique for blind source separation which has been adopted from Ozerov *et al.*

[1] After our research had concluded, we noticed that Hurmalainen *et al.* have performed object-based speaker identification as well [6].

and the improved initialization scheme which has been adopted from Mirzaei *et al.* is discussed as well [2]. These techniques respectively enable enhancement and localization.

In chapter 4, our method for object-based speaker identification is introduced. A technique called automatic relevance determination is presented which has been adopted from Tan *et al.* [9]. This is a method for estimating the model order or the number of objects which shall be used to model a speaker. Additionally, a feature extraction procedure is proposed which is based on group sparsity NMF, a technique which has been developed by Hurmalainen *et al.* [6]. Finally, this chapter concludes with performance results on two data sets; a high quality data set of studio recordings and a data set of field recordings of panel meetings.

In chapter 5, our object-based method for speaker identification is combined with blind source separation as proposed by Ozerov *et al.* and localization as proposed by Mirzaei *et al.* [1, 2]. This chapter concludes with an evaluation of the joint system on two data sets; a non-reverberated data set and a reverberated data set.

Finally, the thesis concludes with a general conclusion in chapter 6.

Chapter 2

State-of-the-art

2.1 Introduction

This chapter aims to give an overview of the state-of-the-art in the field of speaker recognition. Speaker recognition can further be subdivided into speaker verification and speaker identification.

Firstly, **speaker verification**, or in other words speaker authentication, is the task of indicating whether an utterance was spoken by a hypothesized speaker or not. The inputs of the speaker verification system are the acoustic features of the utterance, a hypothesized speaker model and usually a background model containing information about the environment. The output is a boolean label indicating whether a positive match has been found or not.

Secondly, **speaker identification** is the task of labelling an utterance with its corresponding speaker identity obtained from a set of enrolled speakers. In general, speaker identification is considered a more difficult problem than speaker verification. There are two forms of speaker identification [10]. The first approach, *closed-set speaker identification*, chooses a label from a predefined finite set of enrolled speaker identities when given a novel utterance. Note that it returns a label each time. However, this approach may lead to counterintuitive solutions. Suppose a training database contains only male speakers. If a speech segment originating from a female speaker is asked to be identified, closed-set speaker identification will still return a label from the set of male speaker identities. Closed-set indicates that we assume the utterances that need to be labelled are generated by a closed set of known speakers. An extension to closed-set speaker identification additionally returns a ranked list of possible speakers, possibly appended with the corresponding log-likelihood values. The second approach, *open-set speaker identification*, addresses this issue by providing a rejection method. In theory, an open-set speaker identification classifier is able to detect if an utterance is generated by a known speaker. If this is not the case, it will not return a label. Note that open-set speaker recognition can be implemented as a combination of closed-set speaker identification and speaker verification [10].

The remainder of this chapter is organized as follows. Section 2.2 gives a non-exhaustive overview of possible features for speaker recognition. Section 2.3 and 2.4 explain the classic speaker verification and identification system called GMM-UBM. In section 2.5 and section 2.6 support vector machines and i-vectors, two extensions to the GMM-UBM system, are introduced. Even though these two techniques are illustrated in the GMM-UBM framework in these sections, they have other applications in the field of speaker recognition.

2.2 Speaker-dependent features

The goal in speaker recognition is to extract information about the speaker's identity from a speech segment. In order to do so, acoustic features need to be extracted from these segments. As mentioned by Kinnunen *et. al* [11], such features should exhibit sufficiently low intra-class variance and sufficiently high inter-class variance; be robust against noise, distortion or reverberation; not be too difficult to calculate; and not be easy to imitate (for security reasons). The raw data of the speech segment is an unfit candidate for obvious reasons. As a result, a lot of acoustic features have been proposed in the past. This section provides a short, non-exhaustive overview of some frequently used features for speaker recognition.

Firstly, the frequency components obtained from the discrete Fourier transform (DFT) can serve as features. For a frame of N_{FFT} samples the frequency components of the DFT can be calculated as follows

$$v_{freq,fn} = \sum_{t=0}^{N_{FFT}-1} v_n(t) e^{\frac{-j2\pi ft}{N_{FFT}}} \tag{2.1}$$

where f denotes the index of the frequency bin, n denotes the index of the frame, $v_n(t)$ is the t^{th} time domain sample of the n^{th} frame and $v_{freq,fn}$ denotes the frequency component for frequency bin f and frame n. Notice that we denote $v_{freq,fn}$ by a lower case letter, although it is in the frequency domain. We do this because we have the convention to denote scalar values by lower case letters. Following the same convention, $\mathbf{v_{freq,n}}$ is a vector and denotes the entire n^{th} frame in the frequency domain.

Generally the following steps are followed. An incoming stream of auditory information is divided into N frames of N_{FFT} samples. It is common to have an overlap between successive frames. Each of these N frames is subsequently transformed into the frequency domain using the DFT. Since the DFT results in a symmetric frequency spectrum when the time domain signal is real, each frame can fully be represented by $F = \frac{N_{FFT}}{2}$ complex frequency components. The concatenation of all these frames $\mathbf{v_{freq,n}}$ in the frequency domain is called the $F \times N$ complex spectrogram $\mathbf{V_{freq}}$. Usually only the magnitude spectrum $|\mathbf{V_{freq}}|$ is considered because the phase spectrum is assumed to not have any speaker-dependent information. The process of using a sliding window to divide a time-domain signal in frames and subsequently transforming each of these frames into the frequency domain is called the **Short Time Frequency Transform (STFT)**.

Secondly, through frequency warping one may achieve a dimensionality reduction without a severe loss in information. The mel scale is such a frequency warping method based on perceptual, psychoacoustical experiments [12, 13]. Conversion into the mel domain results in the **Mel Frequency Coefficients (MFC)**. Frequency-to-mel conversion can be done with the following formula [14]

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right) = 1127 \log_e\left(1 + \frac{f}{700}\right). \tag{2.2}$$

It is possible to construct a mel filter bank as a set of overlapping triangular filters that span the frequency range. These filters have central frequencies that are evenly spaced in the mel domain but non-uniformly spaced in the frequency domain. In other words, the gap between the central *mels* of adjacent triangular filters of a mel filter bank is the same for the entire range in the mel domain of the incoming signal, but the gap between the central *frequencies* of these

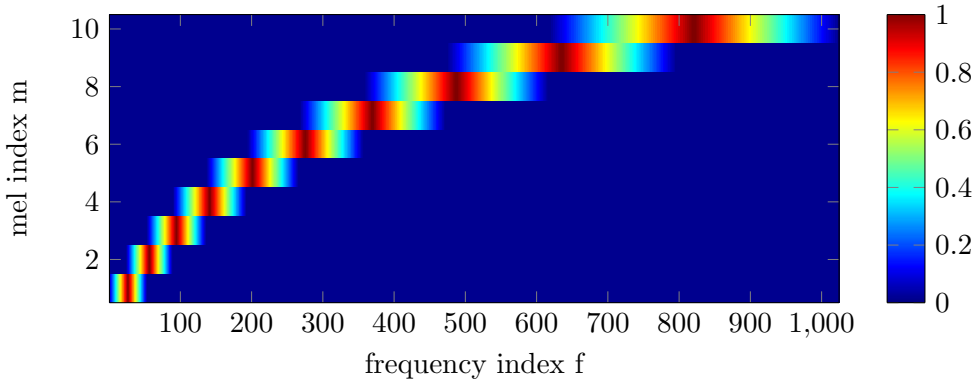


Figure 2.1: The frequency-to-mel transformation matrix $\mathbf{T}$ where F is equal to 1024 and M is equal to 10.

filters will be small at low frequencies and will gradually increase at higher frequencies. The corresponding values of the central frequencies for these evenly spaced mels can be found using the inverse of eq. 2.2. The bandwidth of each triangular filter also increases or decreases as the spacing increases or decreases between adjacent filters so that each frequency is accounted for. In other words, filters located at a low frequency will have a relatively small bandwidth and filters at high frequencies will have a relatively large bandwidth.

If the spectrogram of a segment is available, the output of a mel filter bank with M filters corresponding to that segment is computed with the following matrix product [15]

$$\mathbf{V_{mel}} = \mathbf{T}\,|\mathbf{V_{freq}}| \tag{2.3}$$

where $\mathbf{V_{mel}}$ is the $M \times N$ spectrogram in the mel domain and $|\mathbf{V_{freq}}|$ is the $F \times N$ magnitude spectrogram in the frequency domain. $\mathbf{T}$ is the $M \times F$ frequency-to-mel transformation matrix containing the specification of the M triangular filters [16]

$$t_{mf} = \begin{cases} 0 & \text{for } f_{freq,f} < f_{c,m-1} \\ \frac{f_{freq,f} - f_{c,m-1}}{f_{c,m} - f_{c,m-1}} & \text{for } f_{c,m-1} \leq f_{freq,f} < f_{c,m} \\ \frac{f_{freq,f} - f_{c,m+1}}{f_{c,m} - f_{c,m+1}} & \text{for } f_{c,m} \leq f_{freq,f} < f_{c,m+1} \\ 0 & \text{for } f_{freq,f} \geq f_{c,m+1} \end{cases} \tag{2.4}$$

where $f_{freq,f}$ is the frequency of the f^{th} frequency bin and $f_{c,m}$ is the central frequency of the m^{th} mel filter. In Fig. 2.1 the different filters forming the filter bank are drawn. The coefficients in the n^{th} column $\mathbf{v_{mel,n}}$ of $\mathbf{V_{mel}}$ are the mel frequency coefficients for frame n.

2.3 GMM-UBM for speaker verification

Gaussian Mixture Model - Universal Background Model (GMM-UBM) is a classic method for speaker identification and verification. It was first proposed [17] for speaker identification in a 1990 paper by Rose and Reynolds [18]. Extensions for speaker verification were published in 1995 [19] and 1996 [20]. This technique employs Gaussian mixture models for models of individual speakers as well as a reference model or a background model, as will be explained in the following sections.

Gaussian mixture models are a powerful tool for modeling arbitrary multivariate probability distributions. In GMM-UBM they are used to represent the probability density function for D-dimensional features. There are no constraints on which features are used, as long as they are sufficiently linear and speaker-dependent. A Gaussian mixture model is the weighted linear combination of multiple D-dimensional unimodal Gaussian densities [17]

$$\Pr(\mathbf{x} \mid \boldsymbol{\lambda}) = \sum_{i=1}^{M} w_i \Pr(\mathbf{x} \mid \boldsymbol{\lambda_i}) \tag{2.5}$$

$$\Pr(\mathbf{x} \mid \boldsymbol{\lambda_i}) = \frac{1}{\sqrt{2\pi^D \left|\boldsymbol{\Sigma_i}\right|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu_i})^T(\boldsymbol{\Sigma_i}^{-1})(\mathbf{x}-\boldsymbol{\mu_i})} \tag{2.6}$$

where the parameters of this density are given by $\boldsymbol{\lambda_i} = \{w_i, \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}\}$ which represent the mixture weights w_i, the D-dimensional mean vector $\boldsymbol{\mu_i}$ and the covariance matrix $\boldsymbol{\Sigma_i}$ of size $D \times D$.

For speaker verification, GMM-UBM uses a **likelihood ratio test**

$$\frac{\Pr(\mathbf{x} \mid H_0)}{\Pr(\mathbf{x} \mid H_1)} = \frac{\Pr(\mathbf{x} \mid \boldsymbol{\lambda_{hyp}})}{\Pr(\mathbf{x} \mid \boldsymbol{\lambda_{\overline{hyp}}})} \tag{2.7}$$

$$\Lambda(\mathbf{X}) = \log \Pr(\mathbf{X} \mid \boldsymbol{\lambda_{hyp}}) - \log \Pr(\mathbf{X} \mid \boldsymbol{\lambda_{\overline{hyp}}}) \tag{2.8}$$

where $\mathbf{X}$ is a sequence of feature vectors characterizing the utterance which we wish to verify and $\Lambda(\mathbf{X})$ is the log-likelihood ratio. The likelihood ratio denotes which of two hypotheses is more likely, namely the hypothesis H_0 with parameters $\boldsymbol{\lambda_{hyp}}$ that the utterance belongs to a specific speaker model and the hypothesis H_1 with parameters $\boldsymbol{\lambda_{\overline{hyp}}}$ that the utterance does not belong to that speaker model. If the value of this ratio exceeds a threshold τ, H_0 will be accepted. Otherwise it will be rejected. By applying the logarithm operation, the ratio is converted to an additive expression as in eq. 2.8.

For the latter hypothesis $\boldsymbol{\lambda_{\overline{hyp}}}$, it is necessary to represent all other possible speakers. This is achieved by specifying a **Universal Background Model (UBM)**. There are two major approaches for representing such a UBM: a UBM consisting of a set of speaker models or a UBM consisting of a single model. The latter is the preferred approach since it is more convenient and elegant to handle a single model than handling a set of models. This single model can be constructed by pooling a lot of speaker data together from a sufficient amount of speakers and then fitting a GMM to this probability density.

When constructing a UBM, it is important to take special care of the *composition* of the training data that is used because it will be reflected in the contructed UBM. If there are several subpopulations with e.g. age- or gender-related differences, they should be represented in the training data of the UBM proportionally to the expected composition that will be encountered in the validation or test data. If no prior distribution is known, the data used for constructing the UBM should be uniformly distributed over all subpopulations. Achieving a desired composition in the final UBM can be achieved in different ways. One method consists of pooling the data from several speakers and determining a fitting model. One can also compute a model per subpopulation (for example male and female) and subsequently concatenate both models. This is possible since the Gaussian mixture model is defined by a sum of unimodal gaussian densities. By concatenating both models, we simply define a higher order model with as many unimodal densities as there are in both models for the subpopulations combined. It is then only necessary to renormalize the mixture weights w_i such that they sum to one.

The **Expectation-Maximization (EM) algorithm** [21] is used to fit a model to data from a multitude of speakers, or, in other words, for finding the model parameters $\boldsymbol{\lambda}$. Specifically, it will be used to create a UBM. This EM algorithm can be described as a Maximum Likelihood Estimation approach wherein two separate steps are iterated until the sought after parameters converge. First, in the E-step, the expectation of the log-likelihood under the current model parameters is calculated. Secondly, in the M-step, the parameters are updated by maximizing the expectation of the log-likelihood found in the first step. These iterations continue until the difference in parameters due to an update is sufficiently small. Depending on the data, just five iterations are enough for convergence [17]. This enables a fast model fitting. Additionally, as Reynolds, Quatieri and Dunn state [17], this process can be sped up if the covariance matrix of each unimodal Gaussian is diagonal. Not only does this reduce the parameters for each covariance matrix $\boldsymbol{\Sigma_i}$ by a factor D, it also results in a computationally efficient algorithm since the inverse of a diagonal matrix is simply found by replacing each of its diagonal elements by its inverse. Also, each density modelled by a full-covariance GMM can be modelled by a higher order GMM with reduced covariance matrices $\boldsymbol{\Sigma_i}$ [17].

As previously mentioned, the parameters $\boldsymbol{\lambda}_{\text{UBM}}$ for the UBM are found using the EM algorithm. However, a slightly different approach is needed for modeling individual speakers. **Bayesian Adaptation (BA)** adapts the parameters of the UBM to obtain a specific model for each speaker based on the speaker enrollment data [17]. This is done in a manner similar to EM. In the first step of the iterative algorithm, estimations of the sufficient statistics

$$\begin{aligned}
\Pr(i \mid \mathbf{x_n}) &= \frac{w_i \Pr(\mathbf{x_n} \mid \boldsymbol{\lambda_i})}{\sum_{j=1}^{M} w_j \Pr(\mathbf{x_n} \mid \boldsymbol{\lambda_j})} \\
n_i &= \sum_{t=1}^{T} \Pr(i \mid \mathbf{x_n}) \\
E_i(\mathbf{x}) &= \frac{1}{n_i} \sum_{n=1}^{N} \Pr(i \mid \mathbf{x_n}) \mathbf{x_n} \\
E_i(\mathbf{x}^2) &= \frac{1}{n_i} \sum_{n=1}^{N} \Pr(i \mid \mathbf{x_n}) \mathbf{x_n}^2.
\end{aligned} \tag{2.9}$$

for each mixture i are computed using the previous parameter estimates. This computation is equivalent to computing the expectation of the log-likelihood. These sufficient statistics fully characterize the current estimate of the probability distribution of feature vectors $\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_N}\}$. However in a second step, instead of updating the parameters (or sufficient statistics), as done in the EM algorithm, the updated parameters are combined with their previous value. It is possible to adapt each of the mixture parameters; the weights w_i, the mean vectors $\boldsymbol{\mu_i}$ and the covariance matrices $\boldsymbol{\Sigma_i}$. However, performance does not drop significantly if only the mean vectors $\boldsymbol{\mu_i}$ of each density i are adapted [17]. The update equations are in that case

$$\hat{w}_i = w_i \tag{2.10}$$

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{i}} = \alpha_i^{\boldsymbol{\mu}} E_i(\mathbf{x}) + (1 - \alpha_i^{\boldsymbol{\mu}}) \boldsymbol{\mu_i} \tag{2.11}$$

$$\hat{\boldsymbol{\Sigma}}_{\boldsymbol{i}} = \boldsymbol{\Sigma_i}. \tag{2.12}$$

where $\boldsymbol{\lambda} = \{w_i, \boldsymbol{\mu_i}, \boldsymbol{\Sigma_i}\}$ are the parameter estimations of the previous estimations and $\hat{\boldsymbol{\lambda}} = \{\hat{w}_i, \hat{\boldsymbol{\mu}}_{\boldsymbol{i}}, \hat{\boldsymbol{\Sigma}}_{\boldsymbol{i}}\}$ are their updates for the current iteration.

For each unimodal Gaussian density, a data-dependent adaptation coefficient $\alpha_i^{\boldsymbol{\mu}}$ determines how much the mean vector $\boldsymbol{\mu_i}$ for density i should be affected by the new statistics. This data-dependent adaptation coefficient enables densities i with a high count of data n_i to rely more on the new statistics and reversely keep the old statistics if the count of data n_i is low. This adaptation coefficient is defined as

$$\alpha_i^{\boldsymbol{\mu}} = \frac{n_i}{n_i + r^{\boldsymbol{\mu}}} \tag{2.13}$$

where $r^{\boldsymbol{\mu}}$ is a relevance parameter which determines how much data needs to be observed before a density will be adapted. It is also important to notice that the order (= the amount of unimodal Gaussian densities in the mixture model) for a speaker model is the same as the order of the UBM, since the speaker model is derived from the UBM.

Using models for individual speakers that are adapted from the UBM have some advantages. Firstly, a speaker model which is adapted from the UBM can be stored in a compact way by only storing the mean vectors $\boldsymbol{\mu_i}$ for each unimodal Gaussian density i or by storing the differences of the mean vectors $\boldsymbol{\mu_i}$ with respect to those of the UBM. If features have dimension D and there are M densities in the GMM, the mean vectors will also have dimension D and the adapted speaker model can be stored with complexity $\mathcal{O}(DM)$. Secondly, a feature vector in a large feature space only lies close to a few components of the mixture model. Such components only affect log-likelihood values of features that lie close. Because speaker models are adapted from the UBM, the components of a speaker model still have a correspondence with the components of the UBM [17]. As a result, a fast scoring procedure is possible. Without fast scoring, the evaluation of the log-likelihood ratio needs $2M$ Gaussian computations; M computations for the UBM and M computations for the adapted speaker model [17]. However, the log-likelihood $\Pr(\mathbf{x} \mid \boldsymbol{\lambda_{hyp}})$ can be approximated using only the top C components that correspond to the components of the UBM which contributed most to the log-likelihood $\Pr(\mathbf{x} \mid \boldsymbol{\lambda}_{\overline{\boldsymbol{hyp}}})$. As a result, only $M + C$ computations are necessary for each feature vector [17]. A commonly used value for C is 5, which is typically far less than the full model order M. As a result, the computational complexity is reduced by half. When there are multiple hypothesised speakers, the improvement is even greater [17]. Thirdly, and this has rather to do with robustness than computational speed, the log-likelihood value of a speaker versus the used background model will be unaffected by unseen acoustic events. When training a speaker model, the areas that do not contain any observations will not be adapted and as such be similar to the UBM. If a feature belonging to such an unseen acoustic event appears when scoring, the likelihoods of the background model and the speaker model will cancel each other out.

2.4 GMM-UBM for speaker identification

As indicated in section 2.1, there are two forms of speaker identification. In the GMM-UBM system, closed-set speaker identification can be achieved by searching in the predefined speaker database for the speaker identity which leads to the largest *a posteriori* probability for the

observed feature sequence $\mathbf{X}$:

$$\begin{aligned}\hat{S} &= \arg\max_{S} \Pr(S \mid \mathbf{X}) \\ &= \arg\max_{S} \Pr(\boldsymbol{\lambda}_S \mid \mathbf{X}) \\ &= \arg\max_{S} \frac{\Pr(\mathbf{X} \mid \boldsymbol{\lambda}_S) \Pr(\boldsymbol{\lambda}_S)}{\Pr(\mathbf{X})} \\ &= \arg\max_{S} \Pr(\mathbf{X} \mid \boldsymbol{\lambda}_S) \Pr(\boldsymbol{\lambda}_S).\end{aligned} \tag{2.14}$$

This is the MAP estimate. Between the third and fourth step the term $\Pr(\mathbf{X})$ was neglected since it simply acts as a scale factor and does not influence the maximum value. Similarly, the ML estimate can be obtained by only maximizing over $\Pr(\mathbf{X} \mid \boldsymbol{\lambda}_S)$ when no information about the prior $\Pr(\boldsymbol{\lambda}_S)$ is known. It may be desirable to return several of the most likely speaker labels, possibly appended with the corresponding *a posteriori* probabilities or likelihood values. However, as mentioned in section 2.1 the biggest problem with closed-set speaker identification is that it may lead to misleading results when segments from unknown speakers are encountered.

Open-set speaker identification deals with this problem by returning a label $\varnothing$ when it is expected that the segment belongs to no known speaker. To achieve this, verification is performed on the output $\hat{S}$ of eq. (2.14) (see section 2.3).

2.5 SVM-GSV: a hybrid approach using SVM

Generative models such as the GMM-UBM system described in sections 2.3 and 2.4 try to model the complete distribution of speaker-dependent features. In contrast, *discriminative models* focus on defining a boundary in the feature space [22]. Classification of an instance reduces then to determining on which side of the boundary the feature lies in the feature space.

A **Support Vector Machine (SVM)** is an example of such a discriminative approach [23, 24]. Prior to the training phase, examples of several classes are collected and labelled. If the features are representative for the classes, distinct feature clouds will appear in the feature space that correspond to different speaker classes. The goal of training is to find a boundary separating those feature clouds as good as possible and also to maximize the distance to the features closest to that boundary. The observations closest to the separating hyperplane are called the *support vectors* and the gap separating them is called the *margin*. If the feature clouds are linearly separable, then perfect separation is possible. In that case, the separating gap is called a **hard margin** and this margin contains no features at all. However, two feature sets might not be linearly separable and even when it is possible to linearly separate them, it might not lead to the most robust boundary. In that case, some observations will be allowed to lie inside the separating gap which is now called a **soft margin**. When one or both classes contain outliers, it is fair to assume that these outliers are exceptions. The soft-margin SVM will allow these outliers to lie in the separating gap, albeit at a penalty. Even though the boundary does not separate the training data set perfectly and there are some observations in the margin, it does result in a more robust boundary with a much larger margin.

Support vector machines are computationally interesting. A compact representation of the separating hyperplane is possible by storing the normal vector to it. As mentioned earlier, this is the reason why a SVM can only separate sets that are linearly separable. However, this is not entirely true. Even when the data set is not linearly separable, the input space

can be non-linearly transformed according to a **kernel function** so that both classes are linearly separable in the transformed input space. More information about this non-linear transformation can be found in section 4.5.

A **Support Vector Machine for GMM SuperVectors (SVM-GSV)** is a hybrid speaker verification system that has properties from both generative and discriminative approaches [11, 25]. Similar to the generative GMM-UBM system it uses both a UBM as a reference model that models all possible speakers as well as a Maximum A Posteriori method called Bayesian Adaptation to adapt new speakers from this UBM. However, unlike the GMM-UBM system it does not apply a likelihood ratio test to determine if an utterance was produced by a hypothesized speaker. In the SVM-GSV system, a new segment will be verified by adapting a model based on its feature sequence using the same Bayesian Adaptation that was used for training new speakers. The mean vectors resulting from the adaptation are subsequently stacked into a supervector. These supervectors are finally scored by a SVM that was trained for that specific speaker during the training phase. The scoring of such a supervector consists of a single, fast inner product. For more information about the scoring procedure, see section 4.5.

2.6 i-vectors

As described in section 2.5, the main disadvantage of supervectors is their large dimensionality. One could argue that having more dimensions is useful for the classification. However, the additional information conveyed in extra dimensions may be limited and it also increases the need for more instances during training due to the curse of dimensionality. Even though support vector machines are excellent classifiers for dealing with high dimensionalities, it is preferable to reduce the amount of feature dimensions.

This dimensionality reduction is exactly what the **i-vector technique** tries to achieve. One can decompose the supervector $\mathbf{x}$ [26] as the sum of a speaker-independent and a speaker-dependent term

$$\mathbf{x} = \mathbf{m} + \mathbf{T}\mathbf{x}' \tag{2.15}$$

where $\mathbf{x}$ is the supervector, $\mathbf{m}$ is the speaker-independent mean over all supervectors and $\mathbf{T}\mathbf{x}'$ is the speaker-dependent mean-offset. The deviation $\mathbf{T}\mathbf{x}'$ is the product of a low rank total variability matrix $\mathbf{T}$ and an identity vector $\mathbf{x}'$. The i-vector $\mathbf{x}'$ is the new feature of a speaker utterance and has a far lower dimensionality than the supervector $\mathbf{x}$. The methodology used here assumes that we can identify directions of variability that are speaker-dependent. For the details of training the matrix $\mathbf{T}$, the reader is referred to Dehak [26] and Bahari [27].

2.7 Conclusion

In conclusion, we have gone through some important concepts and terminology in the field of speaker recognition and reviewed several traditional speaker recognition methods as well. A revision of these methods is useful as a frame of reference to compare our speaker recognition method with. It is important to remember the following key issues with the existing speaker recognition methods mentioned in this chapter.

The GMM-UBM system has been put forth as the classic method used in speaker verification, by evaluating the likelihood of two hypotheses, and speaker identification, by evaluating the

likelihood of several speaker models or computing the maximum a posteriori probability. A critical disadvantage of the GMM-UBM system is the need for a universal background model. Not only is it necessary to gather a large amount of data for many different speakers for development of such a background model, but the composition of the background model needs to be adapted to the situation where GMM-UBM shall be applied as well. One can wonder how such a system can be applied if such resources are unavailable.

On a separate note, SVM-GSV, a widely used variant of the GMM-UBM system, employs supervectors based on the Gaussians of the speaker model and the background model. As such, the high dimensionality of the supervectors is a major drawback. The dimensionality can be reduced by extracting i-vectors. However, such i-vectors are computed by evaluating speaker-dependent and speaker-independent variability and this variability is dependent on several factors such as room characteristics, microphone characteristics and the composition of the training data.

Are these techniques the most natural methods for distinguishing speakers? For example, it is unclear how a Gaussian mixture model for a speaker can directly lead to a profound understanding for humans as it does for machines. Human hearing employs cues such as characteristic speaker sounds or tone of voice as ways of categorizing speakers. Although the features used in the GMM-UBM might be comprehensible to humans, the eventual speaker model, i.e. the Gaussian mixture model, is not. Generally, machine learning techniques are able to categorize instances according to information without immediate intuitive meaning. However, it is interesting to think of methods for speaker recognition that do. By now, it should be clear that in this thesis, this is actually what we would like to achieve; to propose a new type of speaker model that is efficient and is based on characterization of speaker-specific sounds.

Chapter 3

Framework

3.1 Introduction

The main goal in this thesis is to perform speaker identification using an object-based speaker model that differs from the GMM-UBM approach discussed in chapter 2. These characteristic objects will be extracted from the magnitude spectrogram $\mathbf{V}$ of a speaker recording by factorizing it using **nonnegative matrix factorization (NMF)**. To construct an algorithm that is better suited to the task, a Bayesian extension of nonnegative matrix factorization will be used which takes into account a model of the data that will be factorized. This model makes the assumption that the objects are Poisson-generated. Such a generative model enables a maximum likelihood approach for finding a solution to the factorization problem. Since we look at identification within a localization and source separation framework, the methods used for localization and blind source separation that will be used will be explained as well in this chapter.

Section 3.2 introduces the basic nonnegative matrix factorization technique. Section 3.3 introduces the gamma-Poisson (GaP) model which will be used throughout this thesis and presents the derivation of a variational Bayesian EM algorithm within this model. Section 3.4 shows how nonnegative matrix factorization can also be used within an EM algorithm for blind source separation and how an intelligent initialization scheme can boost its performance.

3.2 Nonnegative matrix factorization

Suppose we have a large database with instances of multivariate data. The dimension of these instances is F. It is probable that this data will need to be manipulated, in which case it would be advantageous if a reduced representation which sufficiently approximates the original data can be found. Several techniques have been proposed to achieve such a data reduction while retaining most of the useful information. A first example, **Principal Component Analysis (PCA)**, does so by analyzing the eigenvectors of the data and selecting only a finite number of these eigenvectors that correspond to the largest eigenvalues. The contribution of the remaining eigenvectors are assumed to be negligible. Any new data is subsequently decomposed as a linear combination of these principal components. This decomposition captures most of the variance of the original data. Another example of a data reduction technique is **Vector Quantization (VQ)**. Data reduction is attained by clustering the data vectors and choosing one prototype vector per cluster. The feature space is segmented in areas corresponding to such prototype vectors. Each data vector is represented by the prototype vector corresponding to the area it lies

in. However, as Lee and Seung mention, these decompositions are not able to discover various components that are latent, yet characteristic. Such components include e.g. facial features such as eyes, a nose or a mouth when decomposing facial images, or different musical notes and corresponding harmonics when decomposing spectral representations of piano recordings [28]. PCA will look for basis vectors into the eigenvectors of the data and thus ensures an orthogonal basis set. However, these basis vectors may contain negative elements which means that several basis vectors will cancel each other out when linearly combined [28]. It is therefore difficult to give a meaningful interpretation to these basis vectors. VQ on the other hand only assigns one prototype vector per data vector [28]. As a result, patterns are found in the data through clustering, but single instances are not decomposed into meaningful components.

However, **Nonnegative Matrix Factorization (NMF)** is an object-based decomposition technique for multivariate data that focuses on discovering positive and meaningful, but latent, components in a set of data. It was first introduced in 1999 in a paper by Lee and Seung [28] who later also provided computationally efficient algorithms for finding the decomposition matrices $\mathbf{W}$ and $\mathbf{H}$ [29]. Suppose a set of nonnegative data is given that needs to be decomposed. This set contains N different observations of dimension F and it is organised in an $F \times N$ matrix $\mathbf{V}$ with nonnegative entries. The goal of NMF is to factorize the matrix $\mathbf{V}$

$$\mathbf{V}^{\mathrm{F}\times\mathrm{N}} \approx \mathbf{W}^{\mathrm{F}\times\mathrm{K}}\,\mathbf{H}^{\mathrm{K}\times\mathrm{N}} \tag{3.1a}$$

$$\mathbf{v_n}^{\mathrm{F}\times 1} \approx \mathbf{W}^{\mathrm{F}\times\mathrm{K}}\,\mathbf{h_n}^{\mathrm{K}\times 1} \tag{3.1b}$$

into two nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$. As previously mentioned, the individual data vectors $\mathbf{v_n}$, which can be found in the columns of $\mathbf{V}$, can be decomposed as shown in eq. (3.1b). In other words, each data vector $\mathbf{v_n}$ can be approximated by a linear combination of the columns of $\mathbf{W}$ and whose weights are to be found in the vector $\mathbf{h_n}$ [29]. This means that $\mathbf{W}$ contains in its columns $\mathbf{w_k}$ the basis vectors or latent objects that each observation $\mathbf{v_n}$ is composed of. Because of this, $\mathbf{W}$ is called the *dictionary*. The elements h_{kn} of $\mathbf{h_n}$ contain the weights of the linear combination and are thus a measure of the presence of a given object in an observation $\mathbf{v_n}$. It is therefore called the *(time) activation matrix*. In the literature, $\mathbf{W}$ and $\mathbf{H}$ have also respectively been called the template matrix and the excitation matrix as well [30].

Notice that there are no additional constraints on the number of basis vectors K that are used for the factorization. The same data set can be factorized using several values for the **model order** K. However, when the model order is chosen to be very low, not all information will be comprised in the reconstruction. If the model order is subsequently increased, the information increase per additional basis vector will initially be large. However, when a certain data dependent value for the model order is attained, the information increase for additional basis vectors will be negligible. This model order will be called the inherent model order K^*, since it is the order inherent in the data. It is inaccurate to compute a factorization with an order lower than K^* since not all information will be comprised in the factorization. It will also be inefficient to compute a factorization with an order higher than K^* since such a model results in overfitting; the additional objects will not model important trends, but rather noisy and coincidental events without any predictive value. Such a noisy model does not fit new data well generally. For now, it will be assumed that an oracle provides us with the inherent model order K^* for a given data set $\mathbf{V}$. However, this will be an important issue when designing the object-based speaker identification system. For a detailed discussion and proposal of a solution for this model order selection, the reader is referred to section 4.3.

Notice also that NMF is a factorization method because $\mathbf{V}$ is factorized into the product of $\mathbf{W}$ and $\mathbf{H}$. PCA and VQ are factorization techniques as well, since they too can be written as a linear combination such as described in eq. (3.1a) [28]. The difference between these techniques lies in the constraints on $\mathbf{W}$ and $\mathbf{H}$, which lead to different basis vectors [28]. For example, NMF results in purely positive basis vectors which is not necessarily the case for PCA. PCA ensures orthogonality between its basis vectors [31]. VQ ensures that all columns of the activation matrix $\mathbf{H}$ have only one non-zero element which is equal to one. This is equivalent to choosing one basis vectors $\mathbf{w_k}$ for each observation $\mathbf{v}_n$ [28]. As mentioned earlier, this forces the basis vectors to be prototypical.

Now that the goal and characteristics of NMF are clear, it is useful to actually know how to compute $\mathbf{W}$ and $\mathbf{H}$ for a given $\mathbf{V}$. Several iterative algorithms for traditional NMF have been described by Lee and Seung [28, 29] and others [32], but they all share a common property; the goal is to minimize a distance measure between $\mathbf{V}$ and $\mathbf{WH}$. In other words, this distance measure is the cost function that needs to be minimized. There are a number of possible candidates for such a distance measure such as the Euclidean distance (EUC), the Kullback-Leibler divergence (KL) and the Itakura-Saito divergence (IS). These divergences are defined as

$$D_{EUC}(\mathbf{A} \mid\mid \mathbf{B}) = \sum_{ij}(a_{ij} - b_{ij})^2 \tag{3.2a}$$

$$D_{KL}(\mathbf{A} \mid\mid \mathbf{B}) = \sum_{ij}(a_{ij} \log \frac{a_{ij}}{b_{ij}} - a_{ij} + b_{ij}) \tag{3.2b}$$

$$D_{IS}(\mathbf{A} \mid\mid \mathbf{B}) = \sum_{ij}(\frac{a_{ij}}{b_{ij}} - \log \frac{a_{ij}}{b_{ij}} - 1). \tag{3.2c}$$

for two matrices $\mathbf{A}$ and $\mathbf{B}$. Notice that the KL and IS divergence are both Bregman divergences. As a result, they are not truly distance metrics since they do not exhibit symmetry [29, 33].

The basic approach for finding an NMF algorithm is to find update equations that gradually decrease the cost function, i.e. the divergence between $\mathbf{V}$ and $\mathbf{WH}$, when applied iteratively. If such update equations can be found, convergence to a local minimum of the cost function is guaranteed. The chosen cost function determines what set of update rules are used, as can be seen in algorithms 1 to 3 in appendix C.1. For proofs of convergence, the reader is referred to Lee and Seung [29] for EUC-NMF and KL-NMF and to Fevotte [32] for IS-NMF. Fevotte *et al.* have written an insightful paper comparing these update equations as well [34].

The choice of which variant of the NMF algorithm should be used is not arbitrary. The cost function is only justified if the data set $\mathbf{V}$ has some underlying statistical properties. The following statistical models for $\mathbf{V}$ are implied when the Euclidean distance, Kullback-Leibler divergence or Itakura-Saito divergence are used [35, 34]:

$$\text{EUC-NMF:} \qquad v_{fn} \sim \mathcal{N}\left(v_{fn} \mid \sum_{k}^{K} w_{fk}h_{kn}, \sigma^2\right) \tag{3.3a}$$

$$\text{KL-NMF:} \qquad v_{fn} \sim \mathscr{P}\left(v_{fn} \mid \sum_{k}^{K} w_{fk}h_{kn}\right) \tag{3.3b}$$

$$\text{IS-NMF:} \qquad v_{fn} \sim \mathscr{G}\left(v_{fn} \mid a, \frac{a}{\sum_{k}^{K} w_{fk} h_{kn}}\right). \tag{3.3c}$$

In other words, NMF with the Euclidean distance, the Kullback-Leibler divergence and the Itakura-Saito divergence as distance measure between $\mathbf{V}$ and $\mathbf{WH}$ respectively imply a Gaussian, Poisson or Gamma generative model for the data set $\mathbf{V}$.

In this work, we assume that $\mathbf{V}$ follows the generative Poisson model similar to Canny [35], Cemgil [30] and Dikmen *et al.* [36]. Since minimizing the KL divergence coincides with using a maximum likelihood approach for finding $\mathbf{W}$ and $\mathbf{H}$ in a generative Poisson model [35, 34], the KL divergence will be used as a cost function in the remainder of this thesis. The Poisson model will be explained more thoroughly in section 3.3.1.

3.3 Bayesian NMF

In section 3.2, we stated how NMF can be used to discover basis vectors or objects in a dataset $\mathbf{V}$. In this section, it will become clear that additional statistical assumptions about the nature of $\mathbf{V}$ can aid in constructing a better method for discovering the K basis vectors.

3.3.1 The Gamma-Poisson model

A commonly used model for $\mathbf{V}$ is the hierarchical and generative **Gamma-Poisson model** [30, 35, 36], or GaP model in short. In this composite model, $\mathbf{V}$ is assumed to be composed of several components $\mathbf{C_k}$

$$\mathbf{V} = \sum_{k=1}^{K} \mathbf{C_k} \tag{3.4}$$

or

$$v_{fn} = \sum_{k=1}^{K} c_{k,fn}. \tag{3.5}$$

where $c_{k,fn}$, or equivalently $[\mathbf{C_k}]_{fn}$, denotes the $(f,n)^{\text{th}}$ element of component $\mathbf{C_k}$. The notation $\mathbf{C}$ will be used for the set of components $\{\mathbf{C_1}, \mathbf{C_2}, ..., \mathbf{C_K}\}$.

Cemgil [30] states that these components are latent sources that represent the contributions of each of the K basis vectors and thus are not immediately apparent when looking at the data. In the GaP model, these components are assumed to be generated according to a Poisson distribution [36]

$$c_{k,fn} \sim \mathscr{P}\left(c_{k,fn} \mid w_{fk} h_{kn}\right), \quad \mathscr{P}\left(x \mid \lambda\right) = e^{-\lambda} \frac{\lambda^x}{x!} \tag{3.6}$$

In other words, the elements of the k^{th} component $\mathbf{C_k}$ follow a Poisson distribution with for each $(f,n)^{\text{th}}$ element an intensity parameter λ that can be found in $[\mathbf{w_k h_k}]_{k,fn}$, the product of the k^{th} column of the dictionary $\mathbf{W}$ and the k^{th} row of the activation matrix $\mathbf{H}$. Note that each component $\mathbf{C_k}$ has the same dimensions as $\mathbf{V}$. Notice also that since

$$\mathbf{C_k} = \mathbf{w_k h_k}, \tag{3.7}$$

each component is of rank 1 as the set of columns $\mathbf{c_{k,n}}$ of component $\mathbf{C_k}$ can be spanned by a single basis vector, i.e. $\mathbf{w_k}$ [1].

A valid assumption when processing speech recordings from a fixed set of known speakers is that the main spectral characteristics of the phones produced by those speakers are speaker dependent, immutable and thus deterministic. However, the activations of these sounds are stochastic, i.e. it is not known in advance which sounds will appear at what point in time. In addition, these activations are sparse since it is improbable that a lot of objects will concurrently appear. Therefore, since $\mathbf{H}$ is stochastic, the activation matrix is given a prior probability modeled by a gamma distribution

$$\begin{aligned} h_{kn} &\sim \mathscr{G}\left(h_{kn} \mid \alpha, \beta\right) \\ &\sim \left[\beta^{\alpha}\Gamma\left(\alpha\right)\right]^{-1} h_{kn}^{\alpha-1} e^{\left(\frac{-h_{kn}}{\beta}\right)}. \end{aligned} \tag{3.8}$$

A gamma prior has been chosen for the activations h_{kn} for several reasons. Firstly, the gamma distribution can assume multiple forms such as sparse and non-sparse distributions. The amount of sparsity is controlled by an interaction of both hyperparameters. The gamma distribution $\mathscr{G}\left(h_{kn} \mid \alpha, \beta\right)$ has a mean equal to $\alpha\beta$ and a variance equal to $\alpha\beta^2$. If the product of the hyperparameters is large, the mean of the elements drawn from this distribution will be small, i.e. many parameters will lie close to or be equal to zero. If in addition the variance $\alpha\beta^2$ is large, there will be some outliers. Only a few elements will have a large magnitude. This is exactly the behavior of a sparse distribution. Both these conditions are satisfied if α is small and β is large. In that case, the probability density function of the elements h_{kn} is largest near zero. Secondly, both the gamma and the Poisson distribution belong to the family of exponential probability distributions. As a result, the gamma distribution and the Poisson distribution are conjugate. This property simplifies some algorithm derivations [36].

3.3.2 Variational Bayesian Expectation-Maximization (VBEM)

In this section, a maximum likelihood (ML) approach is presented for finding the factorization of $\mathbf{V}$ in the GaP model described in section 3.3.1. The goal of the ML approach is to find the parameters that lead to the largest likelihood. To achieve this, an analytical expression for the likelihood is needed. Unfortunately, the expression for the likelihood contains an intractable integral. A solution to this problem comes in the form of **variational Bayesian techniques**. In short, the expression for the likelihood is intractable because it contains the posterior of some latent sources. If the posterior for the latent sources can be approximated a free distribution $q(\cdot)$ with a manageable analytic form, a tractable approximation of the likelihood can be found. As will be explained in this section, this approximation is a lower bound to the true likelihood. An EM algorithm can subsequently be constructed to maximize this lower bound. Since the lower bound to the likelihood is maximized, the true likelihood is maximized as well.

The derivations presented here closely follow those from Cemgil [30], where priors for both $\mathbf{W}$ and $\mathbf{H}$ are assumed, and Dikmen [36], where only a prior for $\mathbf{H}$ is assumed and $\mathbf{W}$ is treated as deterministic. Dikmen [36] considers only the marginal likelihood $\Pr(\mathbf{V} \mid \mathbf{W})$, where the activations h_{kn} have been integrated out. In this section, a unifying approach is given linking both techniques. In the first part of this section, a general derivation of the ML estimator for both $\mathbf{W}$ and $\mathbf{H}$ combined is given. This part closely follows the derivation from Cemgil [30] and concludes with a formulation of an EM algorithm for finding the ML estimators of both $\mathbf{W}$ and $\mathbf{H}$. In the second part of this section, we show how, with some modifications to the

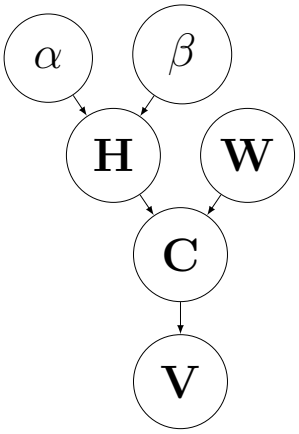


Figure 3.1: Bayesian network of the gamma-Poisson model

derivation from the first part, an EM algorithm can be derived for finding the ML estimator of only $\mathbf{W}$. In this derivation, the contribution of $\mathbf{H}$ is viewed as an additional latent source. Here, $\mathbf{W}$ is considered to be deterministic and $\mathbf{H}$ is considered to be stochastic [36]. This part closely follows the paper from Dikmen [36]. In the third and final part, update equations are derived for the EM algorithm formulated in the second part.

3.3.2.1 Likelihood for W and H

In section 3.3.1, the composite nature of $\mathbf{V}$ was explained. Consider the case of the combined log-likelihood of $\mathbf{W}$ and $\mathbf{H}$ within this GaP model

$$\begin{aligned} \Lambda_{\text{WH}} &= \log \Pr(\mathbf{V} \mid \mathbf{W}, \mathbf{H}) \\ &= \log \Pr(\mathbf{V}, \mathbf{C} \mid \mathbf{W}, \mathbf{H}). \end{aligned} \tag{3.9}$$

In the expression of the log-likelihood in eq. (3.9), the data is augmented with the latent sources $\mathbf{C} = \{\mathbf{C_k}\}$. This step can easily be deducted from the Bayesian network in figure 3.1. These K latent sources $\mathbf{C_k}$ can be marginalized out to obtain an expression for the combined log-likelihood of $\mathbf{W}$ and $\mathbf{H}$

$$\begin{aligned} \Lambda_{\text{WH}} &= \log \sum_{k=1}^{K} \Pr\left(\mathbf{V} \mid \mathbf{C_k}\right) \Pr\left(\mathbf{C_k} \mid \mathbf{W}, \mathbf{H}\right) \\ &= \log \sum_{k=1}^{K} \mathscr{P}\left(\mathbf{V} \mid \mathbf{C_k}\right) = \log \sum_{k=1}^{K} \mathscr{P}\left(\mathbf{V} \mid \mathbf{w_k h_k}\right) \\ &= \log \mathscr{P}\left(\mathbf{V} \mid \sum_{k=1}^{K} \mathbf{w_k h_k}\right) \\ &= \log \prod_{f=1}^{F} \prod_{n=1}^{N} \mathscr{P}\left(v_{fn} \mid \sum_{k=1}^{K} w_{fk} h_{kn}\right) \\ &= \sum_{f=1}^{F} \sum_{n=1}^{N} \log \mathscr{P}\left(v_{fn} \mid \sum_{k=1}^{K} w_{fk} h_{kn}\right). \end{aligned} \tag{3.10}$$

The first step in this derivation is trivial. In the second step, the prior knowledge that the data follows the GaP model is applied. Note that $\mathbf{w_k}$ denotes the k^{th} column of $\mathbf{W}$ and $\mathbf{h_k}$ denotes the k^{th} row of $\mathbf{H}$. The third step is motivated by the superposition principle of Poisson variables which states that, when a random variable is the sum of Poisson random variables with intensity parameters λ_i, then its probability distribution is a Poisson distribution as well with an intensity parameter λ equal to the sum of the intensity parameters λ_i [30]. In the fourth step, the likelihood of $\mathbf{W}$ and $\mathbf{H}$ is rewritten as the product of the likelihoods of the separate elements w_{fk} and h_{kn}.

If we rewrite the Poisson distribution as [30]

$$\begin{aligned}\mathscr{P}(x \mid \lambda) &= e^{-\lambda}\frac{\lambda^x}{x!} \\ &= e^{\log \lambda^x} e^{-\lambda} e^{-\log x!} \\ &= e^{x\log\lambda - \lambda - \Gamma(x+1)}, \quad \Gamma(x) = (x-1)!\end{aligned} \tag{3.11}$$

and substitute this in eq. (3.10), we get the following expression for the combined log-likelihood of $\mathbf{W}$ and $\mathbf{H}$

$$\begin{aligned}\Lambda_{\text{WH}} &= \sum_{f=1}^{F}\sum_{n=1}^{N}\left(v_{fn}\log\sum_{k=1}^{K} w_{fk}h_{kn} - \sum_{k=1}^{K} w_{fk}h_{kn} - \Gamma(v_{fn}+1)\right) \\ &\equiv \sum_{f=1}^{F}\sum_{n=1}^{N}\Big(v_{fn}\log\left[\mathbf{WH}\right]_{fn} - \left[\mathbf{WH}\right]_{fn} - \Gamma(v_{fn}+1)\Big).\end{aligned} \tag{3.12}$$

This is the same expression for the log-likelihood as derived by Cemgil [30].

Now that an expression has been found for the combined log-likelihood of $\mathbf{W}$ and $\mathbf{H}$, an **EM algorithm** can be constructed that maximizes this log-likelihood [30]. The basic approach is to find a lower bound $\Lambda_{\text{WH,LB}}$ on the log-likelihood and find the parameters $\mathbf{W}$ and $\mathbf{H}$ that maximize this bound. If this lower bound is tight enough, in other words if it approximates the true log-likelihood well enough, the $\mathbf{W}$ and $\mathbf{H}$ that are found are also those that lead to the largest true log-likelihood.

The lower bound is

$$\begin{aligned}\Lambda_{\text{WH}} &= \log\Pr(\mathbf{V},\mathbf{C} \mid \mathbf{W},\mathbf{H}) \\ &= \log\sum_{k}^{K}\Pr(\mathbf{V},\mathbf{C_k} \mid \mathbf{W},\mathbf{H}) \\ &= \log\sum_{k}^{K} q(\mathbf{C_k})\frac{\Pr(\mathbf{V},\mathbf{C_k} \mid \mathbf{W},\mathbf{H})}{q(\mathbf{C_k})} \\ &\geq \sum_{k}^{K} q(\mathbf{C_k})\log\frac{\Pr(\mathbf{V},\mathbf{C_k} \mid \mathbf{W},\mathbf{H})}{q(\mathbf{C_k})} = \Lambda_{\text{WH,LB}}\end{aligned} \tag{3.13}$$

Jensen's inequality was applied in the third step of equations (3.13). It can be formulated as $E_q[f(x)] \leq f(E_q[x])$ for any stochastic variable x and distribution $q(x)$,but only if the function $f(\cdot)$ is concave. In this case the function $f(\cdot)$ is the logarithm operation, which is indeed concave, and x is the argument of the logarithm operation. Notice that since $A\log\frac{A}{B} = -A\log\frac{B}{A}$ and

both $q(\mathbf{C_k})$ and $Pr(\mathbf{V}, \mathbf{C_k} \mid \mathbf{W}, \mathbf{H})$ are distributions and thus sum to one, eq. (3.13) indicates nothing more than that $\Lambda_{\text{WH,LB}}$ is equal to the negative of the Kullback-Leibler divergence $D_{KL}(q(\mathbf{C_k}) \mid\mid \Pr(\mathbf{V}, \mathbf{C_k} \mid \mathbf{W}, \mathbf{H}))$ as described in eq. (3.2b).

When $q(\mathbf{C})$ is equal to the posterior of the latent sources $\Pr(\mathbf{C} \mid \mathbf{V}, \mathbf{W}, \mathbf{H})$, then the lower bound $\Lambda_{\text{WH,LB}}$ is equal to the true log-likelihood of $\mathbf{W}$ and $\mathbf{H}$ [30]. Thus,

$$\arg\max_{q(\mathbf{C})} \Lambda_{\text{WH,LB}} = \Pr(\mathbf{C} \mid \mathbf{V}, \mathbf{W}, \mathbf{H}). \tag{3.14}$$

This is the true power of **variational Bayesian methods**. The posterior $\Pr(\mathbf{X} \mid \cdot)$ is approximated by a free distribution $q(\mathbf{X})$ which has a manageable analytical form.

A second step consists of constructing an EM algorithm that uses the lower bound to iteratively maximize the log-likelihood of $\mathbf{W}$ and $\mathbf{H}$ [30]. As previously mentioned, since the lower bound $\Lambda_{\text{WH,LB}}$ is smaller than the log-likelihood, it suffices to maximize $\Lambda_{\text{WH,LB}}$ for $\mathbf{W}$ and $\mathbf{H}$. The maximization step for iteration n thus consists of finding the following estimates for $\mathbf{W}$ and $\mathbf{H}$

$$\begin{aligned}
\left(\mathbf{W}^{(n)}, \mathbf{H}^{(n)}\right) &= \arg\max_{\mathbf{W},\mathbf{H}} \Lambda_{\text{WH,LB}} \\
&= \arg\max_{\mathbf{W},\mathbf{H}} \left(\sum_k^K q(\mathbf{C_k}) \log \frac{\Pr(\mathbf{V}, \mathbf{C_k} \mid \mathbf{W}, \mathbf{H})}{q(\mathbf{C_k})} \right) \\
&= \arg\max_{\mathbf{W},\mathbf{H}} \left(\sum_k^K q(\mathbf{C_k}) \left(\log \Pr(\mathbf{V}, \mathbf{C_k} \mid \mathbf{W}, \mathbf{H}) - \log q(\mathbf{C_k}) \right) \right) \\
&= \arg\max_{\mathbf{W},\mathbf{H}} \left(\sum_k^K q(\mathbf{C_k}) \log \Pr(\mathbf{V}, \mathbf{C_k} \mid \mathbf{W}, \mathbf{H}) \right).
\end{aligned} \tag{3.15}$$

To find a solution for eq. (3.15) the distribution $q(\mathbf{C})$ needs to be known. As mentioned earlier, the lower bound is tight when $q(\mathbf{C})$ approaches the posterior of the latent sources $\Pr(\mathbf{C} \mid \mathbf{V}, \mathbf{W}, \mathbf{H})$. The solution to this problem is to approximate $q(\mathbf{C})$ by the posterior $\Pr(\mathbf{C} \mid \mathbf{V}, \mathbf{W}, \mathbf{H})$ for the estimates $\left(\mathbf{W}^{(n-1)}, \mathbf{H}^{(n-1)}\right)$ computed in the previous estimation [30]. Notice that the E-step in any EM algorithm consists of computing the sufficient statistics. In this case, this is equivalent to finding a $q(\mathbf{C})$ that sufficiently approximates the posterior $\Pr(\mathbf{C} \mid \mathbf{V}, \mathbf{W}, \mathbf{H})$.

The EM algorithm for finding the maximum likelihood estimates of $\mathbf{W}$ and $\mathbf{H}$ can be summarized as follows

$$\boxed{\begin{aligned}
&\text{E step} \quad q(\mathbf{C}) = \Pr(\mathbf{C} \mid \mathbf{V}, \mathbf{W}^{(n-1)}, \mathbf{H}^{(n-1)}) \\
&\text{M step} \quad \left(\mathbf{W}^{(n)}, \mathbf{H}^{(n)}\right) = \arg\max_{\mathbf{W},\mathbf{H}} \left(\sum_k^K q(\mathbf{C_k}) \log \Pr(\mathbf{V}, \mathbf{C_k} | \mathbf{W}^{(n-1)}, \mathbf{H}^{(n-1)}) \right)
\end{aligned}} \tag{3.16}$$

A similar derivation of this EM algorithm can also be found in [30].

3.3.2.2 Likelihood for W

The EM algorithm in (3.16) serves to find both $\mathbf{W}$ and $\mathbf{H}$ concurrently. However, in this thesis the goal is to extract solely the basis vectors in $\mathbf{W}$ because these will serve as a speaker model.

Any temporal information found in $\mathbf{H}$ is assumed irrelevant for identification in this setting[1]. It is therefore that from now on, we will only work with the marginal log-likelihood Λ_{W} where $\mathbf{H}$ has been marginalized out. This paradigm is equivalent to treating $\mathbf{H}$ as an additional latent source. As a result, a few modifications to the more general derivation that was previously presented are needed. This will lead to a slightly different EM algorithm.

Firstly, we do not wish to approximate the posterior $\Pr(\mathbf{C} \mid \mathbf{V}, \mathbf{W}, \mathbf{H})$ by $q(\mathbf{C})$. Instead, the posterior $\Pr(\mathbf{C}, \mathbf{H} \mid \mathbf{V}, \mathbf{W})$ will be approximated by the instrumental distribution $q(\mathbf{C}, \mathbf{H})$ [36]. Note that, in this case, the log-likelihood and its lower bound are defined as

$$\Lambda_{\mathrm{W}} = \log \Pr(\mathbf{V}, \mathbf{C}, \mathbf{H} \mid \mathbf{W}) \tag{3.17}$$

and

$$\Lambda_{\mathrm{W,LB}} = \sum_{k}^{K} q(\mathbf{C_k}, \mathbf{H}) \log \frac{\Pr(\mathbf{V}, \mathbf{C_k}, \mathbf{H} \mid \mathbf{W})}{q(\mathbf{C_k}, \mathbf{H})} \tag{3.18}$$

where the lower bound is tightest when $q(\mathbf{C_k}, \mathbf{H})$ is equal to the posterior $\Pr(\mathbf{C_k}, \mathbf{H} \mid \mathbf{V}, \mathbf{W})$. Be sure to compare these equations with eqs. (3.9) and (3.13). As previously mentioned, $\mathbf{H}$ is no longer treated as a parameter, but as a latent source.

Secondly, since

$$\begin{aligned} \log \Pr(\mathbf{C}, \mathbf{H} \mid \mathbf{V}, \mathbf{W}) &= \log \Pr(\mathbf{V} \mid \mathbf{W}) \Pr(\mathbf{C}, \mathbf{H} \mid \mathbf{W}) \\ &= \log \Pr(\mathbf{V} \mid \mathbf{W}) + \log \Pr(\mathbf{C}, \mathbf{H} \mid \mathbf{W}) \end{aligned} \tag{3.19}$$

and $\Pr(\mathbf{V} \mid \mathbf{W})$ does not depend on $\mathbf{C}$ and $\mathbf{H}$, it suffices to approximate $q(\mathbf{C}, \mathbf{H})$ by $\Pr(\mathbf{C}, \mathbf{H} \mid \mathbf{W})$ [36]. The EM algorithm now takes on the following form

$$\boxed{\begin{aligned} &\text{E step} \quad q(\mathbf{C}, \mathbf{H}) = \Pr(\mathbf{C}, \mathbf{H} \mid \mathbf{W}^{(n-1)}) \\ &\text{M step} \quad \mathbf{W}^{(n)} = \arg\max_{\mathbf{W}} \left(\sum_{k}^{K} q(\mathbf{C_k}, \mathbf{H}) \log \Pr(\mathbf{V}, \mathbf{C_k}, \mathbf{H} \mid \mathbf{W}^{(n-1)}) \right). \end{aligned}} \tag{3.20}$$

3.3.2.3 Update equations

First we have shown how an EM algorithm can be derived for finding both $\mathbf{W}$ and $\mathbf{H}$ concurrently. Subsequently, we have shown how some modifications to this more general derivation, i.e. by treating $\mathbf{H}$ as a latent source as well, leads to an EM algorithm for finding only $\mathbf{W}$. The main difference between these two derivations is the following; in the former, the posterior $\Pr(\mathbf{C} \mid \mathbf{W}, \mathbf{H})$ is modelled by the free distribution $q(\mathbf{C})$, and in the latter, the posterior $\Pr(\mathbf{C}, \mathbf{H} \mid \mathbf{W})$ is modelled by the free distribution $q(\mathbf{C}, \mathbf{H})$. The major advantage of using a free distribution to model the posterior is that it has a manageable analytic form. However, nothing has yet been said about the characteristics of this free distribution $q(\mathbf{C}, \mathbf{H})$. Here, we shall show what the most natural choice of distribution for $q(\mathbf{C}, \mathbf{H})$ is. Given this, the specific update equations for the EM algorithm shown in eq. (3.20) will be derived.

[1] Although in this thesis temporal information is not considered, it would be an interesting extension. A possible way to do so within the NMF framework is by using exemplars similar to the works of Gemmeke *et al.* [37, 5]

The first step in deriving update equations for the EM algorithm in (3.20) is finding the sufficient statistics for $q(\mathbf{C}, \mathbf{H})$ necessary for the expectation step. First, an analytic representation for $q(\mathbf{C}, \mathbf{H})$ must be chosen. Therefore, the **mean field approximation** [38, 39], which states that the free distribution $q(\mathbf{C}, \mathbf{H})$ is completely factorized over the latent variables $\mathbf{C}$ and $\mathbf{H}$, is applied. We can thus write

$$q(\mathbf{C}, \mathbf{H}) \approx q(\mathbf{C})q(\mathbf{H}). \tag{3.21}$$

Consider the vector $\mathbf{c_{fn}} = \{c_{1,fn}, c_{2,fn}, ...c_{K,fn}\}$ for a given time frame n and frequency bin f. This vector contains the K component elements $c_{k,fn}$ corresponding to every object for that time-frequency bin. Dikmen states that the posterior for this vector follows a multinomial distribution with corresponding probabilities $p_{k,fn}$ [36]. Since the posterior for $\mathbf{c_{fn}}$ is multinomial, the most natural choice for $q(\mathbf{c_{fn}})$ is a multinomial distribution as well. As a result, the following expression is obtained for the free distribution $q(\mathbf{C})$ modeling the posterior of the components

$$\begin{aligned} q(\mathbf{C}) &= \prod_{f=1}^{F}\prod_{n=1}^{N} q(\mathbf{c_{fn}}) \\ &= \prod_{f=1}^{F}\prod_{n=1}^{N} \mathscr{M}(c_{1,fn}, ..., c_{K,fn} \mid p_{1,fn}, ..., p_{K,fn}) \\ &= \prod_{f=1}^{F}\prod_{n=1}^{N} \frac{\Gamma\left(\sum_{k=1}^{K} c_{k,fn} + 1\right)}{\prod_{k=1}^{K}\Gamma\left(c_{k,fn}+1\right)} \prod_{k=1}^{K} (p_{k,fn})^{c_{k},fn}. \end{aligned} \tag{3.22}$$

A similar reasoning can be made for the free distribution $q(\mathbf{H})$ [36]. The most appropriate choice as a distribution for the activations is the gamma distributions [30, 36]. As a result, the following expression is obtained for the free distribution $q(\mathbf{C})$ modelling the posterior of the activations

$$\begin{aligned} q(\mathbf{H}) &= \prod_{k=1}^{K}\prod_{n=1}^{N} q(h_{kn}) \\ &= \prod_{k=1}^{K}\prod_{n=1}^{N} \mathscr{G}(h_{kn} \mid \alpha, \beta) \end{aligned} \tag{3.23}$$

where the gamma distribution $\mathscr{G}(\cdot \mid \cdot)$ is specified in Eq. (3.8).

Cemgil [30] states that an optimum can be found for each of the disjoint distributions $q_{\mathbf{x}}$ where $\mathbf{x} \in \{\mathbf{c_{fn}}, h_{kn}\}$ if

$$q_{\mathbf{x}} = e^{\langle \log p(\mathbf{C},\mathbf{H}|\mathbf{W})\rangle_{q/q_{\mathbf{x}}}} \tag{3.24}$$

where $\langle\cdot\rangle_q$ is the expectation with respect to distribution q. The distribution $q/q_{\mathbf{x}}$ denotes the distribution $q(\mathbf{C}, \mathbf{H})$ without distribution $q_{\mathbf{x}}$ where $\mathbf{x} \in \{\mathbf{c_{fn}}, h_{kn}\}$. Applied to the set of latent variables, this leads to the following update equation for $q(\mathbf{c_{fn}})$ found by Cemgil [30]

$$\begin{aligned} q(\mathbf{c_{fn}}) &= q(\{c_{1,fn}, ..., c_{K,fn}\}) \\ &= e^{\langle \log p(\mathbf{C},\mathbf{H}|\mathbf{W})\rangle_{q(\mathbf{C},\mathbf{H})/q(\mathbf{c_{fn}})}} \\ &= e^{\sum_{k}^{K}\left(c_{k,fn}\log w_{fk} + c_{k,fn}\langle\log h_{kn}\rangle - \log\Gamma(c_{k,fn}+1) + c\right)}. \end{aligned} \tag{3.25}$$

In eq. (3.25), c is a constant. As also stated by Cemgil [30], the probabilities for the multinomial distributions $q(\mathbf{c_{fn}})$ are equal to

$$p_{k,fn} = \frac{w_{fk} e^{\langle \log h_{kn} \rangle}}{\sum_{k'}^{K} w_{fk'} e^{\langle \log h_{kn} \rangle}}. \tag{3.26}$$

Since the probability distribution for the activations h_{kn} is equal to the gamma distribution, the following expression is found for $\langle \log h_{kn} \rangle$

$$\langle \log h_{kn} \rangle = \psi(\alpha_{kn}) + \log(\beta_{kn}) \tag{3.27}$$

where $\psi(x) = \frac{d}{dx} \log \Gamma(x)$. This is nothing more than the logarithmic mean of the gamma distribution.

Using the same method as for $q(\mathbf{c_{fn}})$, Dikmen [36] and Cemgil [30] have found the following update equation for $q(h_{kn})$

$$\begin{aligned} q(h_{kn}) &= e^{\langle \log p(\mathbf{C},\mathbf{H}|\mathbf{W}) \rangle_{q(\mathbf{C},\mathbf{H})/q(h_{kn})}} \\ &= e^{-\left(\frac{\alpha_{kn}}{\beta_{kn}} + \sum_{f}^{F} w_{fk}\right) h_{kn} + \left(\alpha_{kn} - 1 + \sum_{f}^{F} \langle c_{k,fn} \rangle\right) + c} \end{aligned} \tag{3.28}$$

where $\langle c_{k,fn} \rangle = p_{k,fn} v_{fn}$ and c is a constant. The gamma distribution is to be recognized in this expression for $q(h_{kn})$ where the hyperparameters are equal to

$$\begin{aligned} \alpha_{kn} &= \alpha_{kn} + \sum_{f}^{F} \langle c_{k,fn} \rangle \\ \beta_{kn} &= \left(\frac{\alpha_{kn}}{\beta_{kn}} + \sum_{f}^{F} w_{fk} \right). \end{aligned} \tag{3.29}$$

All the necessary sufficient statistics of $q(\mathbf{C})$ and $q(\mathbf{H})$ for calculating the expectation needed in the E-step of eq. (3.20) are now available. The next task is to maximize the objective function in the **M-step** from eq. (3.20) to obtain an estimate for $\mathbf{W}$. This leads to the following simple update equation for the elements of the dictionary [36]

$$w_{fk} = \frac{\sum_{n} \langle c_{k,fn} \rangle}{\sum_{n} \langle h_{kn} \rangle}. \tag{3.30}$$

3.4 Object-based blind source separation

Blind source separation is an unsupervised signal processing technique for discovering and extracting individual source signals from a mixture signal [40]. Blind source separation is used in e.g. the field of neuroscience when analyzing EEG signals. Each EEG electrode produces a signal recording brainwaves at a particular location on the patient's scalp. The goal of blind source separation is to estimate physiologically meaningful active areas within the brain producing these brain waves with nothing but the information from the EEG electrodes [40].

Blind source separation has also successfully been applied on separation of recordings of several speakers and musical recordings [1, 41]. Instead of EEG electrodes, the recording elements are microphones and the active areas correspond to auditory sources.

The objective of blind source separation can be stated as follows. Consider a set of recording microphones, which shall be referred to as receivers, that record the signals originating from several sound sources, which shall be referred to as sources. In this section, $\mathbf{V}$ denotes the $F \times N \times I$ *complex* mixture spectrogram which contains the I *complex* channel mixture spectrograms $\mathbf{V_i^c}$ corresponding to the I receivers. Each of the $F \times N$ *complex* source spectrograms $\mathbf{V_s}$ contributes to each *complex* channel spectrogram $\mathbf{V_i^c}$. The contribution of the S individual source spectrograms $\mathbf{V_s}$ to each channel mixture spectrogram $\mathbf{V_i^c}$ can be modeled by channel coefficients $a_{is,f}$, contained in the $I \times S \times F$ mixing matrix $\mathbf{A}$. The mixing matrix acts as a transfer function between source and receiver. In other words,

$$\mathbf{v_{i,n}^c} = \sum_{s}^{S} \mathbf{a_{is}} \cdot \mathbf{v_{s,n}} \tag{3.31}$$

where $\mathbf{a_{is}}$ is a $F \times 1$ column vector containing the complex channel coefficients from source s to receiver i and $\mathbf{v_{i,n}^c}$ and $\mathbf{v_{s,n}}$ are the n^{th} frames of the complex channel spectrogram $\mathbf{V_i^c}$, as recorded by receiver i, and the complex source spectrogram $\mathbf{V_s}$ respectively. In other words, each row of $\mathbf{V_s}$ corresponds to a certain frequency bin and is multiplied by its corresponding channel coefficient $a_{is,f}$. The contributions of each source are subsequently summed to obtain the complex channel spectrogram $\mathbf{V_i^c}$. Simply put, the goal of blind source separation is to retrieve all original source spectrograms $\mathbf{V_s}$ if the complex channel spectrograms $\mathbf{V_i^c}$ are known.

An early paper by Sajda *et al.*, exploring the possibilities of nonnegative matrix factorization for blind source separation, attempted to recover original images from a mixture of underlying spectral images [42]. Sajda states that NMF performs the separation several orders of magnitude faster than existing source separation techniques at the time [42]. Hopefully, the fast nature of the NMF algorithm will enable real-time solutions for source separation in the near future, opening up a broad range of applications in the fields of bio-metrics and signal processing. In this thesis, our focus lies in the field of two-channel blind source separation of audio recordings.

3.4.1 Monaural versus multichannel object-based blind source separation

The first distinction to be made is between blind source separation of monaural signals and blind source separation of multichannel signals. The former is considered a more difficult task, since in this case spatial information is unavailable. Wang performed such one-channel source separation on music signals by decomposing the magnitude spectrogram into components using regular unsupervised NMF and subsequently grouping objects originating from one source under supervision [43]. At this point, only the magnitude spectrogram $|\mathbf{V_s}|$ for each individual source s is retrieved. The phase spectrogram is retrieved by applying a separate mask for each source on the phase spectrogram of the original mixture signal. The mask denotes whether a source is dominant for each time-frequency bin. Notice that this technique does not perform actual blind source separation, since the grouping of the components into clusters corresponding to a single source is done under supervision.

However, Virtanen achieved true monaural blind source separation using a variant of NMF which, on the one hand, enforces sparse solutions and, on the other hand, adds an extra term to the cost function favoring solutions exhibiting temporal continuity [41]. This extra term

specifically penalizes the changes in gain for consecutive frames. However, such a technique primarily works well in the case of pitched instruments, since these show a large amount of temporal continuity [41]. Audio recordings of e.g. drum beats have a turbulent nature and thus show rapid changes in magnitude over consecutive frames. The condition of temporal continuity is unfulfilled in such a case. It is therefore that blind source separation of multichannel signals will generally lead to better results.

One problem that occurs when dealing with such multichannel data is that the NMF algorithm in its original form is unsuited for dealing with multiple streams of data [1]. Extensions towards multichannel NMF have been studied several times. Possible solutions include a simple stacking, where the different channel spectrograms are combined into a single data matrix $\mathbf{V}$ [44], and nonnegative tensor factorization, where the data is structured into a tensor and subsequently factorized [45, 46]. As Ozerov *et al.* state, these techniques do not account for convolutive mixing [1]. This causes problems for real-world applications where reverberation is an issue. To account for this, Ozerov *et al.* have derived a blind source separation technique that can deal with convolutive mixtures [1]. More information about this technique is given below.

3.4.2 Object-based multichannel blind source separation: a maximum likelihood approach using EM

Ozerov *et al.* have developed a very interesting multichannel blind audio source separation method relying on a maximum likelihood EM algorithm which maximizes the joint likelihood of the multichannel data [1]. Furthermore, a model is used that can deal with both convolutive mixing and a limited amount of stationary and spatially uncorrelated noise. This model is equal to the model stated in eq. (3.31) but for an additional term which takes into account spatially uncorrelated noise. For a given time-frequency bin, this model can be stated as follows:

$$\mathbf{v_{fn}^c} = \mathbf{A_f}\mathbf{v_{fn}} + \mathbf{b_{fn}} \tag{3.32}$$

or in full

$$\begin{bmatrix} v_{1,fn}^c \\ \dots \\ v_{I,fn}^c \end{bmatrix} = \begin{bmatrix} a_{11,f} & \dots & a_{1S,f} \\ \dots & \ddots & \vdots \\ a_{I1,f} & \dots & a_{IS,f} \end{bmatrix} \times \begin{bmatrix} v_{1,fn} \\ \dots \\ \dots \\ v_{S,fn} \end{bmatrix} + \begin{bmatrix} b_{1,fn} \\ \dots \\ b_{I,fn} \end{bmatrix}. \tag{3.33}$$

Notice the difference between $\mathbf{v_{fn}^c}$ at the left-hand side of the equation and $\mathbf{v_{fn}}$ at the right-hand side of the equation. The term $\mathbf{v_{fn}^c}$ refers to the vector containing the different channels of the complex mixture spectrogram for a given time-frequency bin. The term $\mathbf{v_{fn}}$ refers to the set of *complex* source spectrograms for a given time-frequency bin.

This convolutive mixture model is valid when the channel coefficients in $\mathbf{A_f}$, or in other words the frequency domain impulse response from the sources to the receivers, do not change in time. When there is a temporal change of the impulse response with time, which could for example be due to a change in location of one of the sources, then the mixing matrix should depend on the time index n as well.

Ozerov *et al.* define a model wherein they assume each complex source spectrogram to be a sum of mutually and individually independent complex components:

$$v_{s,fn} = \sum_{k \in K_s} c_{k,fn} \quad , c_{k,fn} \sim \mathcal{N}_c\left(0, w_{s,fk} h_{s,kn}\right) \tag{3.34a}$$

which is identical to

$$v_{s,fn} \sim \mathcal{N}\left(0, \sum_{k \in K_s} w^s_{fk} h^s_{kn}\right) \tag{3.34b}$$

where $\mathcal{N}_c(\mu, \sigma)$ denotes the proper complex Gaussian distribution and $K_s = \{...\}$ is the set of component indices belonging to source s. Notice that these components are *complex*, unlike the components previously defined in the GaP model. Within the model from eq. (3.34), the mixture model can also be expressed in terms of the complex components $\mathbf{c_{fn}}$

$$\mathbf{v^c_{fn}} = \mathbf{\mathring{A}_f c_{fn}} + \mathbf{b_{fn}} \tag{3.35}$$

or in full

$$\begin{bmatrix} v_{1,fn} \\ ... \\ v_{I,fn} \end{bmatrix} = \begin{bmatrix} \mathring{a}_{11,f} & \mathring{a}_{12,f} & ... & ... & \mathring{a}_{1K,f} \\ \mathring{a}_{21,f} & \mathring{a}_{22,f} & ... & ... & \mathring{a}_{2K,f} \\ ... & ... & \ddots & & \vdots \\ ... & ... & & \ddots & \vdots \\ \mathring{a}_{I1,f} & \mathring{a}_{I2,f} & ... & ... & \mathring{a}_{IK,f} \end{bmatrix} \times \begin{bmatrix} c_{1,fn} \\ c_{2,fn} \\ ... \\ ... \\ c_{K,fn} \end{bmatrix} + \begin{bmatrix} b_{1,fn} \\ ... \\ b_{I,fn} \end{bmatrix}. \tag{3.36}$$

$\mathbf{\mathring{A}_f}$ is called the augmented matrix and is easily obtainable from the original mixing matrix. Each column of $\mathbf{\mathring{A}_f}$ corresponds to a complex component $\mathbf{C_k}$ from one of the sources. The value of the complex channel coefficients in that column can be found in the column of the original mixing matrix $\mathbf{A_f}$ corresponding to that source. Notice that unlike in the previous section, components can belong to different sources, i.e. speakers, here.

Within this model, Ozerov *et al.* propose a novel blind source separation framework based on NMF. Without going into too much detail, the EM algorithm maximizes the likelihood of the set of parameters $\boldsymbol{\theta} = \{\mathbf{A}, \mathbf{W}, \mathbf{H}, \mathbf{\Sigma_b}\}$ where $\mathbf{A}$ is the $I \times S \times F$ mixing matrix and $\mathbf{\Sigma_b}$ is the $I \times I$ noise covariance matrix [1]. In this framework, the dictionary $\mathbf{W}$ and the activation matrix $\mathbf{H}$ are not the factorization results of the magnitude spectrogram, but the factorization results of the *power* spectrogram. Also, $\mathbf{W}$ contain the objects for all speakers and $\mathbf{H}$ contain activations for all speakers. An additional output of the source separation algorithm indicates to which source an object belongs.

Unfortunately, the generative model used by Ozerov *et al.* is not the GaP model as described in section 3.3.1[2]. Instead, Ozerov *et al.* assume that the components $\mathbf{C}_k$ are mutually and individually independent over time and frequency as specified in eq. (3.34) [1]. This means that for each component, no additional information about a component element $c_{k,fn}$ can be extracted from knowing component elements $c_{k,f'n'}$ corresponding to other time-frequency bins. This assumption is equivalent to a statistical model of superimposed Gaussian components, as shown in eq. (3.34). It can be shown that ML estimation of $\mathbf{W_s}$ and $\mathbf{H_s}$ within such a model

[2]When we use variational Bayesian NMF within the GaP generative model alongside this algorithm based on mutually independent Gaussians, how can we then justify the use of two different generative models for $\mathbf{V}$? In their paper, Ozerov *et al.* state that for their experiments they have initialized their source separation algorithm using KL-NMF [1]. These factorizations are less prone to lower-energy residual artifacts and interferences, unlike IS-NMF factorizations, due to the scale-invariance of the IS divergence [1]. As will be explained later on in this thesis, the variational Bayesian NMF algorithm will have a similar purpose, i.e. to provide an initialization for $\mathbf{W}$ and $\mathbf{H}$. In conclusion, the use of two different generative models is justified.

coincides with minimization of the IS divergence between the power source spectrogram and the reconstruction $\mathbf{W_s H_s}$ [1]:

$$\Lambda_{\text{WH}} = -D_{IS}(|\mathbf{V_s}|^{.2}||\mathbf{W_s H_s}). \tag{3.37}$$

The watchful reader might have noticed that the model of superimposed Gaussians does not coincide with the gamma generative model corresponding to the IS divergence from eq. (3.3c). However, Fevotte has shown that the Gaussian model from eq. (3.3c) is equivalent to the gamma generative model from equation (3.3c) where $a = 1$ [34].

As stated by Fevotte in an earlier paper, the ML estimation of the parameters $\mathbf{W}$ and $\mathbf{H}$ reduces to estimation of the variance parameters within the model stated above [32]. This simplifies the derivation of the EM algorithm [1]. The reader is referred to this paper for a full derivation and update equations of the blind source separation algorithm.

3.4.3 An improved initialization scheme

Ozerov *et al.* and Mirzaei *et al.* state that the EM algorithm developed by Ozerov *et al.* is very sensitive to initialization of the parameters $\boldsymbol{\theta}$ [1, 2]. It is therefore that Ozerov *et al.* propose several initialization schemes, both supervised and unsupervised [1]. However, supervised initialization methods for the parameters will not be considered here since these ruin the purpose of blind source separation. One of the unsupervised initialization methods consists of computing the factorization matrices $\mathbf{W}$ and $\mathbf{H}$ from the separated signals resulting from other source separation techniques such as those from Arberet and Sawada [1]. One can argue, however, that such an initialization is not truly unsupervised since an additional source separation method needs to be fully executed during the initialization process. Another unsupervised initialization scheme proposed by Ozerov *et al.*, which has been specifically applied to blind separation of professionally produced music recordings, consists of concatenating both channel spectrograms and factorizing the concatenation using regular NMF as if it were a monaural spectrogram [1]. The resulting dictionary $\mathbf{W}$ and activation matrix $\mathbf{H}$ are then the initial estimates for the EM algorithm. The channel coefficient estimates are derived from the components [1].

Mirzaei proposes a different initialization scheme for the parameters $\mathbf{A}$, $\mathbf{W}$ and $\mathbf{H}$. In short, the initialization scheme consists of three steps. First, the spatial information that resides within the multichannel recordings is used for estimating the mixing matrix $\mathbf{A}$ and simultaneously counting the number of sources. Secondly, once $\mathbf{A}$ is known, initial source spectrogram estimates are made using a technique called binary masking. Finally, each of these source spectrogram estimates are factorized and the resulting dictionary and activation matrix are used as initialization for the source separation algorithm. Below, each of these three steps is explained more thoroughly.

In the first step, the angles of arrival are estimated for an arbitrary number of sources. Within this initialization scheme, we assume that the microphone array has two microphones, i.e. the number of channels I is equal to 2. Furthermore, we assume that the microphone array is relatively small, i.e. the distance between the microphones is about 0.15 meters. If a sound source is located at a certain bearing relative to the end-fire direction of the microphone array, and this sound source is located sufficiently far from the microphone array [3], there will be a

[3]This ensures that the far-field assumption is satisfied. If this is the case, between-channel delays will only depend on the bearing angle of the sound source location relative to the microphone array.

delay present which only depends on the delay between both channels. If for now we assume that only one sound source is present and only noise is present which is spatially uncorrelated [4], the model can be stated as follows [47]

$$\begin{aligned} v_1^c(t) &= v_1(t) + b_1(t) \\ v_2^c(t) &= \alpha v_1(t+\tau) + b_2(t). \end{aligned} \tag{3.38}$$

where $v_1^c(t)$ and $v_2^c(t)$ are both time-domain signals corresponding to the two microphones of the microphone array, $v_1(t)$ is the source signal and $b_1(t)$ and $b_2(t)$ are the spatially uncorrelated noise signals. Notice the similarities between the model stated here and the model stated in eq. (3.36). However, the model from eq. (3.38) is in the time domain and only has one sound source for now, the model in eq. (3.36) is in the frequency domain and accounts for multiple sound sources.

The goal is to estimate the delay τ. After all, if the time delay τ is known, the bearing angle corresponding to that sound source is known. A straightforward way of computing the delay τ is by evaluating the cross-correlation

$$r_{v_1 v_2}(\tau) = \int_{-\infty}^{+\infty} v_1^c(t) v_2^c(t+\tau) \mathrm{d}t. \tag{3.39}$$

The cross-correlation can also be computed in the frequency domain using the cross power spectral density $g_{v_2 v_2}(f)$ [47]

$$r_{v_1 v_2}(\tau) = \int_{-\infty}^{+\infty} g_{v_1 v_2}(f) e^{j2\pi f \tau} \mathrm{d}f. \tag{3.40}$$

Notice that the cross power spectral density $g_{xx}(f)$ is equal to the power spectral density of the signal x. In practice, we need to estimate the cross power spectral density because the estimates need to be computed on windows of limited duration. If the channel spectrograms $\mathbf{V}_1$ and $\mathbf{V}_2$ are available, the matrix $\hat{\mathbf{G}}_{\mathbf{v_1 v_2}}$ from eq. (3.41) contains in its columns discrete estimates of the cross power spectral density per time frame.

$$\hat{\mathbf{G}}_{\mathbf{v_1 v_2}} = \mathbf{V_1^c} \cdot \mathbf{V_2^{c*}} \tag{3.41}$$

Notice that to calculate the n^{th} frame of $\hat{\mathbf{G}}_{\mathbf{v_1 v_2}}$, only the n^{th} frames of $\mathbf{V_1^c}$ and $\mathbf{V_2^c}$ are needed. As a result, $\hat{\mathbf{G}}_{\mathbf{v_1 v_2}}$ can be updated frame-wise. Similarly, per-frame estimates of the crosscorrelation can be computed by applying eq. (3.41), but only summing over a finite amount of frequency bins resulting from a STFT. The estimates are combined in a matrix $\hat{\mathbf{R}}_{\mathbf{v_1 v_2}}$.

The time delay between both channels within a frame can now be estimated by evaluating the cross power spectral density for a frame, subsequently computing a short-time estimate of the cross-correlation within that frame and finding the value of τ that leads to the largest peak in this estimate of the cross-correlation as follows

$$\tau^* = \arg\max_{\tau} \mathbf{r}_{\mathbf{v_1 v_2, n}}. \tag{3.42}$$

where $\mathbf{r}_{\mathbf{v_1 v_2, n}}$ is the n^{th} frame of $\hat{\mathbf{R}}_{\mathbf{v_1 v_2}}$ and τ^* is the time delay corresponding to a sound source.

[4] If the noise was spatially correlated, it would appear as an additional sound source.

However, as Knapp *et al.* state, the accuracy of the time delay estimate may be improved by pre-filtering both channel signals [47]. As such, if we denote the pre-filters by $h_1(f)$ and $h_2(f)$, the generalized cross-correlation is defined as:

$$\begin{aligned} r_{v_1'v_2'}(\tau) &= \int_{-\infty}^{+\infty} h_1(f)h_2^*(f)g_{v_1v_2}(f)e^{j2\pi f\tau}\mathrm{d}f \\ &= \int_{-\infty}^{+\infty} \psi_G(f)g_{v_1v_2}(f)e^{j2\pi f\tau}\mathrm{d}f \end{aligned} \tag{3.43}$$

where v_1' and v_2' denote the pre-filtered microphone signals and $\psi_G(f)$ denotes the chosen frequency weighting. Several generalized cross-correlation metrics are possible with different frequency weightings $\psi_G(f)$ [47].

One of the proposed generalized cross-correlation metrics proposed by Knapp *et al.* is the **Generalized Cross Correlation with Phase Transform (GCC-PHAT) metric**. Frequency weighting corresponds here to normalizing the cross power spectral density $g_{v_1v_2}(f)$ to unity amplitude:

$$\psi_G(f) = \frac{1}{\mid g_{v_1v_2}(f) \mid} \tag{3.44}$$

As a result, the amplitude of the cross power spectral density of the pre-filtered signals is equal to one for all frequencies. Only the phase of the cross power spectral density is used to localize a sound source. If we apply this frequency weighting function $\psi_G(f)$ to the short-time estimates of the crosspower spectral density $\hat{\mathbf{G}}_{\mathbf{v_1v_2}}$, the GCC-PHAT metric can be used as follows

$$\hat{r}_{fn\theta} = \mathrm{Real}\left(\frac{v_{1,fn}^c v_{2,fn}^{c*}}{|v_{1,fn}^c v_{2,fn}^{c*}|} e^{\frac{j2\pi df_{freq,f}\cos(\theta)}{c}}\right) \tag{3.45a}$$

$$m_{fn\theta} = 1 - arctan(\alpha\sqrt{(1-\hat{r}_{fn\theta})}) \tag{3.45b}$$

$$a_{spec,\theta} = \max_n \sum_f^F m_{fn\theta}. \tag{3.45c}$$

where $v_{2,fn}^{c*}$ is the element-wise complex conjugate of $v_{2,fn}^c$. In eqs. (3.45), $v_{1,fn}^c$ and $v_{2,fn}^c$ are the time-frequency bins of the channel spectrograms $\mathbf{V_1^c}$ and $\mathbf{V_2^c}$. The parameters d, c and $f_{freq,f}$ denote the distance between the microphones, the velocity of sound and the frequency corresponding to frequency bin f respectively. The symbol $\hat{r}_{fn\theta}$ denotes an element of the $F \times N \times A$ matrix $\hat{\mathbf{R}}$. The parameter A denotes the number of angles at which $\hat{\mathbf{R}}$ is being evaluated. The symbol $m_{fn\theta}$ is an element of the matrix $\mathbf{M}$, which has the same dimensionality as $\hat{\mathbf{R}}$. $\mathbf{M}$ is an element-wise non-linear transformation of $\hat{\mathbf{R}}$. Mirzaei *et al.* state that such a non-linear transformation is useful for sharpening peaks that correspond to angles of true sound sources [2]. A good choice for the non-linear parameter is $\alpha = 3$ [2].

The $A \times 1$ column vector $\mathbf{a_{spec}}$ has elements $a_{spec,\theta}$ and is called the **angular spectrum**. It enables counting and localization of the sources. As mentioned earlier, if a physical source is present at a certain angle relative to the microphone array, a peak will appear in the angular spectrum at that angle θ_s where s is the source index. A peak finding algorithm is applied to identify and count these peaks. The amount of peaks will hopefully be equal to the amount of speakers simultaneously present in the recording. However, special care needs to be taken when detecting peaks in the angular spectrum in the case of reverberated settings. Reflections can

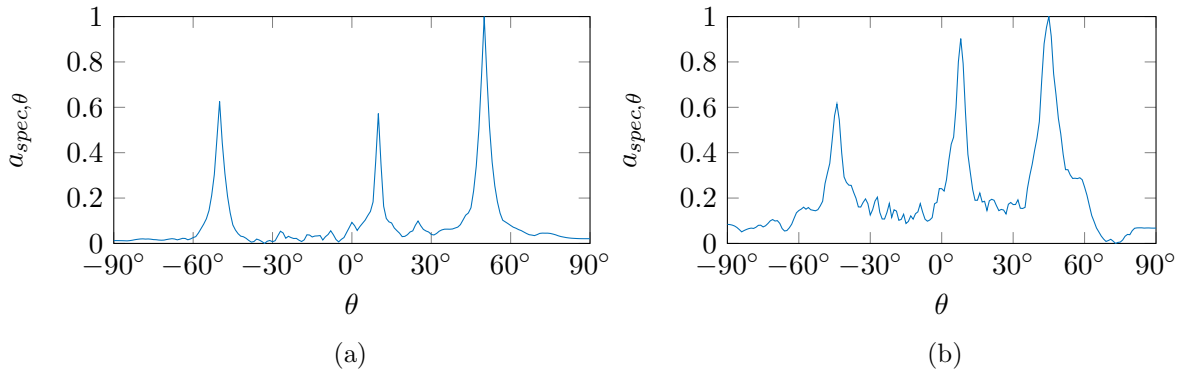


Figure 3.2: Angular spectrum $\mathbf{a_{spec}}$ for the (a) non-reverberated and (b) reverberated case. Speakers are located at angles −50°, 10° and 50°. The range of the angular spectrum has been normalized to $[0, 1]$

lead to additional peaks in the angular spectrum. If these additional peaks are associated to sources, a wrong estimate of the count of the sources will be made. Knapp *et al.* state that the GCC-PHAT metric is rather robust for reverberated signals. In figure 3.2, two angular spectra are shown for the instantaneous mixing case and the reverberated case. Additional conditions for the height of the peaks and separation between the peaks in the angular spectrum can avoid such erroneous source detections.

The initial estimate of the mixing matrix for source s can now be computed as follows

$$\mathbf{A_{INIT,i,s}} = \mathbf{1^{F\times 1}} \qquad , i = 1 \tag{3.46}$$

$$\mathbf{A_{INIT,i,s}} = e^{\frac{j2\pi d \mathbf{f_{freq}} \cos(\theta_s)}{c}} \qquad , i = 2. \tag{3.47}$$

When the initial estimate of the mixing matrix $\mathbf{A_{INIT}}$ has been computed, it can be used for the second step.

In the second step, the source spectrograms $\mathbf{V_s}$ of each speaker are estimated using a technique called **binary masking**. First, each time-frequency bin is evaluated to see which source is dominant in it. For each source spectrogram estimate $\mathbf{V_s^{BM}}$, only those bins where that source is dominant are set to the mixture spectrogram and all other bins are set to zero. Determining whether a source is dominant in a time-frequency bin is done using the same metric used in the counting and localization step. Only those elements of $\mathbf{M}$ that correspond to angles θ_s where a source was detected are considered. For each time frequency bin, speaker s is considered dominant if the corresponding element $m_{fn\theta_s}$ is largest. Binary masking can be stated as follows

$$\begin{aligned} i_{BM,fn} &= \max_s m_{fn\theta_s} & s &= 1..S \\ v_{s,fn}^{BM} &= v_{1,fn}^{c} & i_{BM,fn} &= s \\ v_{s,fn}^{BM} &= 0 & i_{BM,fn} &\neq s \end{aligned} \tag{3.48}$$

where $v_{s,fn}^{BM}$ is an element of the initial estimate for the source spectrogram $\mathbf{V_s^{BM}}$ of speaker s and $i_{BM,fn}$ indicates which source is dominant for a given time-frequency bin.

In the third step, the initial estimates of the dictionary $\mathbf{W}$ and the activation matrix $\mathbf{H}$ are computed with the Bayesian nonnegative matrix factorization method demonstrated in section 3.3.

All initial estimates are now available that are needed for Ozerov's EM algorithm. Mirzaei has shown that this initialization scheme greatly improves the performance [2]. Furthermore, we would like to mention that we have made modifications to the algorithms provided by Ozerov *et al.* and Mirzaei *et al.*. We have tried to vectorize as many functions as possible in the source separation algorithm. This speeds up the source separation algorithm significantly.

3.5 Conclusion

In this chapter, the key concept has been nonnegative matrix factorization. This object-based technique enables extraction of positive objects $\mathbf{w_k}$ which are characteristic to a set of data observations $\mathbf{V}$. Besides the traditional algorithms for nonnegative matrix factorization, which are based on different cost functions, we have introduced a Bayesian variant which estimates model parameters with Bayesian inference assuming the generative gamma-Poisson model. Furthermore, we have shown how nonnegative matrix factorization can be applied in blind source separation and how an improved initialization scheme including speaker localization can improve the performance of source separation.

In this thesis, the data set $\mathbf{V}$ is a magnitude spectrogram and the objects $\mathbf{w_k}$ are sounds that are characteristic to a speaker. At the end of the previous chapter, we raised the question if it is possible to build a speaker model that is based on speaker-specific sounds. Thanks to nonnegative matrix factorization, these speaker specific sounds can be extracted from a speaker spectrogram $\mathbf{V_s}$. As a result, $\mathbf{W_s}$ can serve as a speaker model.

However, some problems remain. First, it is unclear how the correct inherent model order K^* can be found based on data. Secondly, there is no direct way of comparing different speaker dictionaries $\mathbf{W_s}$ with each other. A method for feature extraction from a speech fragment will need to be specified in order to identify speech segments. In the next chapter, these issues will be addressed.

Chapter 4

Object-based Speaker Recognition

4.1 Introduction

In Chapter 2, speaker recognition with GMM-UBM has been discussed, as it has been the standard speaker recognition method for the past few decades. In summary, GMM-UBM is a generative model which models the distribution of feature vectors based on training data and uses a background model to compare speaker models against.

Recently, object-based speech processing techniques have been proposed as an alternative method for several different applications such as speech separation [1], automatic music transcription [4] and speech recognition [6]. The main approach consists of extracting phone-like objects $\mathbf{w_{s,k}}$ from a speaker spectrogram $\mathbf{V_s}$ via nonnegative matrix factorization. Speech-related information can subsequently be extracted from these objects.

In this chapter, we propose a novel method for speaker recognition which uses the dictionary $\mathbf{W_s}$ as a speaker model. The dictionary is obtained using nonnegative matrix factorization using variational Bayesian inference as explained in chapter 3. Furthermore, group sparsity NMF is proposed as a means for extracting features from a set of frames based on a set of speaker dictionaries $\mathbf{W_{TOT}}$. Nonnegative matrix factorization with a group sparsity constraint has already been applied by Hurmalainen *et al.* to speech recognition [6]. We show here that it can also be applied to speaker identification with competitive performance compared to other speaker identification methods[1]. The greatest advantage of the proposed technique is that the development of a universal background is no longer necessary. Also, sufficiently accurate identification is possible in the order of 0.5 seconds. To the best of our knowledge, object-based speaker recognition has only been studied by Joder *et al.* and their method differs from our ours [7].

In order for the method described here to be applicable, some conditions need to be satisfied. To sum up, the following assumptions are made throughout this chapter:

- Only monaural audio signals occur. As a result, blind source separation is infeasible. Therefore, we assume that speakers do not speak simultaneously.

[1] After our research had concluded, we noticed that Hurmalainen *et al.* do perform speaker identification in a group sparsity model as well. However, our methods do differ. Hurmalainen *et al.* use exemplars instead of regular objects. As a result, our speaker dictionaries have a lower dimensionality, i.e. K is typically between 10 and 20 in our method and Hurmalainen *et al.* use 250 exemplars of 34 atoms per speaker or 8500 atoms per speaker. Although our method differs, we feel obliged to mention our mistake and we sincerely apologize for any mention of originality about the idea for using group sparsity NMF in speaker identification in the sections that follow.

- Only utterances from a closed set of known speakers in the conversation. Hence, speaker verification is not needed. The focus lies entirely on speaker identification.

The remainder of this chapter is organized as follows. Section 4.2 gives the design of the object-based speaker recognition system. The full system is broken down into components with a well-defined purpose. Sections 4.3 to 4.5 elaborate upon important aspects of the speaker recognition system; section 4.3 introduces an improved method for estimating model orders, section 4.4 explains how GS-NMF can be used for feature extraction and section 4.5 describes how support vector machines can be used for multiclass classification. This chapter concludes with an evaluation of object-based speaker recognition on several data sets in section 4.6. In this final section, a comparison is given between our method and speaker recognition as performed by Joder *et al.* as well [7].

4.2 Design of a speaker identification system

In this section, the general framework of the proposed object-based speaker identification is given. While designing the system, we have tried to keep everything modular in order to keep a clear structure. Without going into too much detail, each of these modules is explained concisely. Sections 4.3 to 4.5 elaborate on some important steps; automatic relevance determination, feature extraction with group sparsity NMF and multiclass classification with support vector machines.

As with most machine learning techniques, speaker identification can be divided into two main parts. First, a **learning phase**, or training phase, is initiated. In this phase, the models are built. Second, during the **identification phase**, new instances are identified using the models trained previously during the learning phase. It is important that the two sets of instances used for training and validation are disjoint. Otherwise, the accuracy of the identification system might be overestimated since some instances occur in the identification phase that have already been encountered during training.

Figures 4.1 and 4.2 contain schematics of both the learning and identification phases.

4.2.1 Learning phase

In figure 4.1 an overview is given of the learning phase of the speaker identification system. In short, the goal is to build a separate model for each speaker by extracting a speaker dictionary $\mathbf{W_s}$ based on recordings of that speaker. These speaker dictionaries will enable feature extraction. Finally, a classifier will need to be trained using labeled features.

4.2.1.1 Pre-Processing: Selection and STFT

The first step in building a model for a speaker is the selection of time-domain audio recordings from a database. This might seem an obvious step, but it is a crucial step nonetheless. If any noise is present in the recordings the model is built upon, such as a noisy car driving by or a baby crying, it will be included in the model for that speaker. Any reappearance of such noise in the identification phase will wrongfully be identified as the speaker corresponding to the corrupted model, which is evidently undesirable. The **Pre-Processing** step begins with the selection of adequate audio segments. When a sufficient amount of time-domain signals has been gathered, these signals need to be converted into the frequency domain using short-time spectral analysis. A sliding window is applied to the time-domain signaly to divide it into

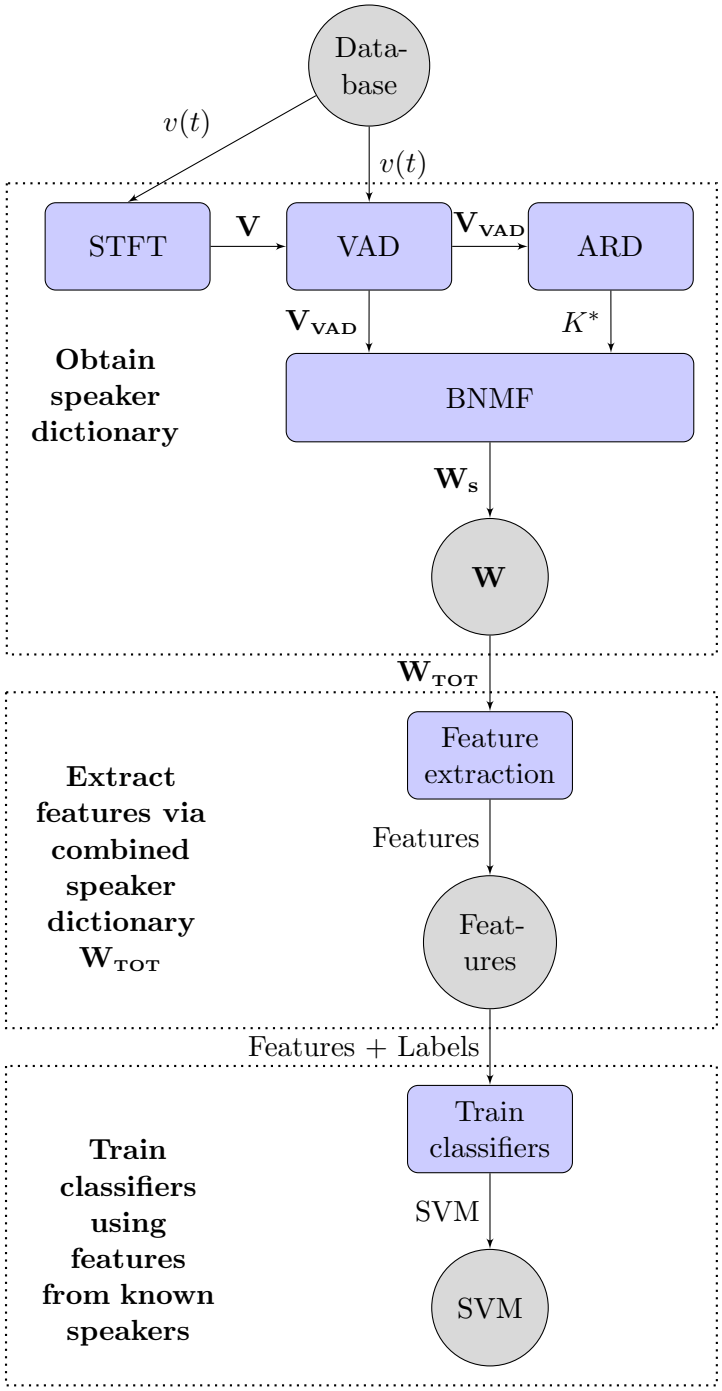


Figure 4.1: Learning phase of object-based speaker identification

individual frames. Each of these frames is subsequently transformed into the frequency domain using the STFT to obtain the spectrogram. The window length N_{FFT} of the sliding window denotes the amount of samples each frame contains. It is common to use a power of two for the window length because in that case the *Fast Fourier transform algorithm* can be applied. The size of the window length is a trade-off between temporal and frequency resolution. We assume

that the human articulators for speech production do not change significantly within a window frame of approximately 50 milliseconds. For instance, if the data has a sample frequency of 16 kHz, we choose a window length N_{FFT} of 1024 samples which corresponds to a temporal resolution of $64ms$. For identification purposes, only the amplitude of the spectrum is useful. Furthermore, since we deal with real signals in the time domain, the amplitude spectrum is symmetric and the amount of frequency bins can be reduced by half without loss of information. If the window length N_{FFT} is equal to 1024, there are 512 frequency bins, corresponding to a frequency resolution of $\frac{8000\text{Hz}}{512} = 15.625\text{Hz}$. The amount of frequency bins is denoted by the parameter F. The overlap of adjacent frames is a parameter that needs to be chosen as well. Here, it is set to 50%, which corresponds to an overlap of 512 samples or 32 milliseconds. The resulting spectrogram $\mathbf{V}$ has dimensions $F \times N$, where F is equal to 512 and N is the amount of frames. Notice that we have used the parameter F for the dimension of the observations $\mathbf{v_n}$ as well. Indeed, the spectrogram $\mathbf{V}$ is the matrix to be factorized and its frames are the observations $\mathbf{v_n}$.

4.2.1.2 Voice activity detection (VAD)

Performing speaker identification on frames which are void of speech is superfluous. Therefore, some sort of speech detection should be implemented to avoid redundant computations. The most basic speech detection method is energy-based. The average energy of the n^{th} frame is calculated as follows

$$E_n = \frac{1}{N_{FFT}} \sum_{t=(n-1).N_{FFT}+1}^{n.N_{FFT}} v^2(t) \tag{4.1}$$

where $v(t)$ is the time-domain signal, n is the frame index and N_{FFT} is the window length in samples [48]. Notice that E_n is always larger than or equal to 0 and smaller than or equal to 1, independent of window length, because $v(t)$ is always between -1 and 1. This energy-based feature is compared with a threshold E_{tresh} in order to label a frame as speech or silence. The threshold may be fixed or computed adaptively based on the data. Here, only little to no noise is assumed to be present, so a fixed threshold is sufficient. An appropriate choice for the threshold E_{tresh} is 10^{-6}. However, other techniques with better performance such as adaptive energy-based VAD [48], zero-crossing rate VAD or a fusion of several VAD methods [49] can be used as well to improve the performance of VAD.

The frames that correspond to non-speech are simply disregarded, i.e. the frame is assigned a label $\varnothing$. The speech frames are passed on to the next block: automatic relevance determination, i.e. model order selection.

4.2.1.3 Automatic relevance determination (ARD)

When factorizing a data set $\mathbf{V}$, the model order K is the amount of objects, i.e. basis vectors, that are used to decompose $\mathbf{V}$. It is the number of columns of $\mathbf{W}$, as well as the amount of rows in $\mathbf{H}$. As already mentioned in chapter 3, there is no direct way of determining the correct inherent order K^* of a data set $\mathbf{V}$.

A possible solution consists of evaluating the likelihood of the factorization for a range of values for K. Initially, the likelihood will increase for solutions with increasing model orders. However, as soon as the inherent model order K^* is attained, the likelihood will stagnate. Determining the right model order thus reduces to finding the knee point in the graph of the

model order versus the likelihood. Such techniques have been applied by Cemgil [30] and Mirzaei *et al.* [50, 2]. The main disadvantage of this method is that a factorization needs to be computed for several values of the model order K, making it computationally expensive. Therefore, a technique developed by Tan *et al.* is adopted here where model order estimation is computed concurrently with the factorization [9]. This technique is called automatic relevance determination[2], and will be discussed thoroughly in section 4.3. The obtained inherent model order K^* is passed on to the next block: BNMF.

4.2.1.4 Nonnegative matrix factorization with Bayesian inference (BNMF)

Once an appropriate model order has been found, objects can be extracted from the data through NMF with Bayesian inference as explained in section 3.3. This is the core element of the object-based speaker identification system. The elements of the time activation matrix $\mathbf{H}$ are irrelevant for identification: only the non-temporal characteristic objects in the speaker dictionaries $\mathbf{W_S}$ are considered relevant here[3]. The objects of the decomposition included in $\mathbf{W_S}$ are stored so they can be accessed by the next block: feature extraction.

4.2.1.5 Feature extraction

The general procedure in machine learning is to train a classifier based on some features extracted from instances. Defining these features is non-trivial. Speaker dictionaries are obtained through BNMF. However, these dictionaries are far too high-dimensional to be used as features. To illustrate this, assume that a speaker can be characterized by 15 objects that are each composed of 512 frequency bin values. In this case, a classifier would need to be built in a 7680-dimensional feature space. Building conventional classifiers for such features would take an enormous amount of time.

Therefore, another kind of feature with fewer dimensions needs to be extracted based on the speaker dictionaries. The feature extraction is done using **group sparsity nonnegative matrix factorisation (GS-NMF)**, which is discussed in more detail in section 4.4. For this approach, the combined dictionary $\mathbf{W_{TOT}}$ is necessary. This combined dictionary contains every objects from all known speakers. Thus, dictionaries need to be learned in advance for all speakers before features can be extracted.

As will be shown later, the approach in this work is to group several adjacent time frames into blocks and subsequently extract one feature vector for each block of frames.

[2] We would like to add that there is still room for improvement in estimating the model order for a given speaker. Automatic relevance determination is done concurrently with a factorization, but this factorization isn't used in this system. Instead, only the obtained inherent model order is used to perform BNMF. However, instead of performing ARD and BNMF sequentially, an algorithm could be derived which combines both. For more information, we would like to refer the reader to section 4.3 and the final conclusion in chapter 6 for more information.

[3] Various other implementations of NMF exist which take into account temporal information. For example, Smaragdis uses a convolutive extension of the regular NMF algorithm for supervised speech separation [51]. Such a decomposition takes into account dependencies across the columns of $\mathbf{V}$ [51]. In other words, temporal information is stored as well in the dictionary of the final decomposition. The same approach can be applied for speaker identification since the decomposed objects, even if they are now composed of several successive frames, are speaker-dependent as well. Another possible method for incorporating temporal information into a nonnegative decomposition consists of performing NMF on *exemplars* instead of frames [52]. An exemplar is nothing more than a concatenation of multiple successive frames.

4.2.1.6 Classifier training

Once the features for several speakers have been extracted, a classifier can be trained. This classifier divides the feature space in several regions corresponding to speakers. A good classifier ensures that the boundaries lie as far away as possible from single-class feature clouds corresponding to different speakers. Several classifiers are possible[4], but we have used support vector machines in this work.

Naturally, support vector machines (SVM) are binary classifiers that divide a feature space in two regions. In section 4.5, we explain how binary SVMs can be used for multiclass classification.

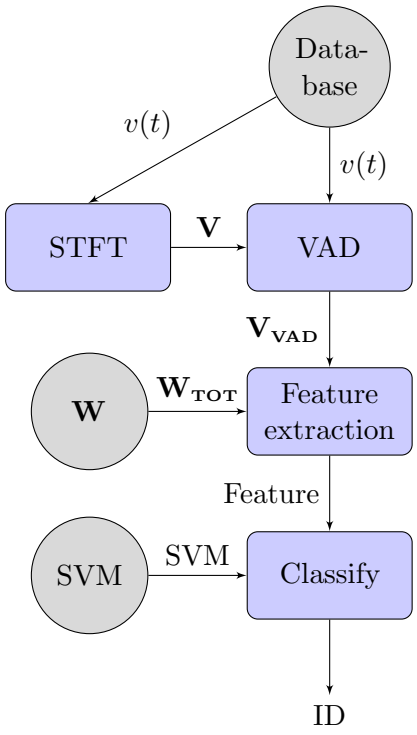


Figure 4.2: Identification phase of object-based speaker identification

4.2.2 Identification phase

The diagram of the identification phase is given in figure 4.2. First, an incoming audio stream needs to be transformed into the frequency domain using STFT. The silent frames from the resulting spectrogram are subsequently disregarded. The remaining spectrogram is divided into blocks and a feature vector is extracted for each block, based on the combined speaker dictionary $\mathbf{W_{TOT}}$. Finally, features are categorized using the support vector machines trained during the learning phase.

Indeed, the identification phase is quite similar to the learning phase. All steps mentioned above, except for classification, are identical to their corresponding counterparts in the learning phase. However, classifiers no longer need to be trained as they have been built during the learning phase.

[4]The most appropriate classifiers are support vector machines and artificial neural networks. These have been applied extensively in the literature.

Both automatic relevance determination and Bayesian nonnegative matrix factorization are unnecessary during identification since the models for the speakers no longer need to be learned. As a consequence, the actual speaker identification is computationally far less intensive than learning speaker models.

In this section, no further information will be given about the intermediate steps as they have been briefly explained in section 4.2.1. Feature extraction and classification will be discussed in-depth in section 4.4 and section 4.5 respectively.

4.3 Automatic relevance determination

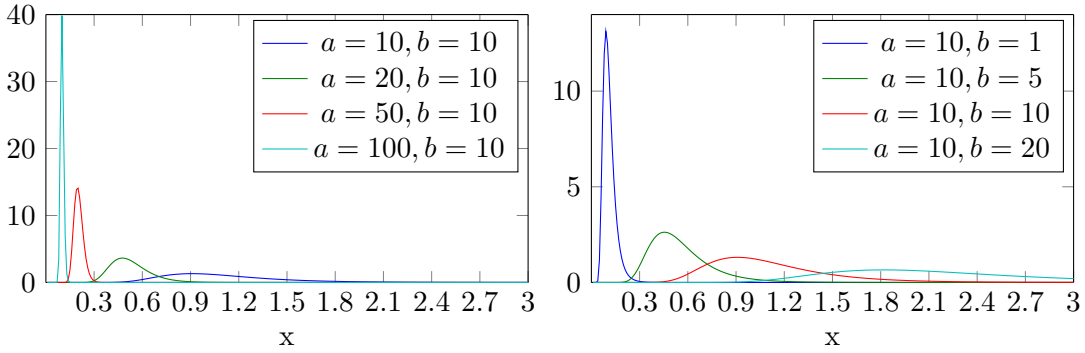
In Chapter 3 and particularly section 3.2, NMF has been introduced as a method for low-rank decomposition. In this work, the set of observations $\mathbf{V}$ to be decomposed is the magnitude spectrogram. The decomposition is equivalent to writing each of the observations $\mathbf{v_n}$ as a linear combination of K basis vectors $\mathbf{w_k}$ which are contained in the dictionary $\mathbf{W}$. However, no method is provided for finding the correct model order K in ordinary nonnegative matrix factorization. The problem of finding the latent dimensionality of the subspace spanned by the basis vectors can be solved using a technique called automatic relevance determination, which has been proposed by Tan *et al.* [9]. This technique has been adopted from a similar problem statement for principal component analysis (PCA), where the model order is being estimated automatically using Bayesian inference as well [53]. Tan *et al.* have extended this technique to Bayesian NMF for several divergence functions, corresponding to different assumed statistical models for the data $\mathbf{V}$ (see section 3.2). The focus in this section lies on maximizing the Kullback-Leibler divergence between the original and reconstructed data for reasons explained in section 3.3.1.

The basic approach for ARD is to initiate NMF with an initial order K_{init} which is sufficiently large, and subsequently prune away irrelevant objects, thus decreasing the model order until no irrelevant objects are left. At that point, the inherent model order of the data is attained. In order to prune away irrelevant objects, Tan *et al.* have introduced **relevance weights** λ_k which denote the importance of component $\mathbf{C_k}$ [9]. The k^{th} object $\mathbf{w_k}$, corresponding to component $\mathbf{C_k}$, is specified in the k^{th} column of the dictionary $\mathbf{W}$ and has corresponding weights $\mathbf{h_k}$ in the k^{th} row of the activation matrix $\mathbf{H}$. This means that each relevance weight needs to be tied to a column of $\mathbf{W}$ and a row of $\mathbf{H}$. One possible way of doing so is by specifying in the Bayesian model that elements of $\mathbf{w_k}$ and $\mathbf{h_k}$ are directly generated from the relevance parameters λ_k. Given the relevance parameter λ_k, Tan *et al.* define the following generative model for the elements w_{fk} and h_{kn}

$$\begin{aligned} p(w_{fk}|\lambda_k) &= \frac{1}{\lambda_k} e^{-\frac{w_{fk}}{\lambda_k}} \\ p(h_{kn}|\lambda_k) &= \frac{1}{\lambda_k} e^{-\frac{h_{kn}}{\lambda_k}}. \end{aligned} \tag{4.2}$$

These probability distributions are nothing more than the exponential distributions with rate parameter equal to $\frac{1}{\lambda_k}$ [9]. The relevance parameter λ_k acts as a variance-like parameter [9]. When λ_k is small, the parameters w_{fk} and h_{kn} will very likely be equal to or near zero and vice versa. Consequently, a small relevance weight indicates that an object has low importance and that it can be neglected without affecting the factorization significantly.

The relevance weights are given prior distributions as well [9]



(a) (b)

Figure 4.3: The inverse-gamma distribution for several settings of the hyperparameters; (a) several settings for a and (b) several settings for b

$$\begin{aligned} p(\lambda_k|a,b) &= \mathcal{IG}(\lambda_k|a,b) \\ &= \frac{b^a}{\Gamma(a)}\lambda_k^{-a-1}e^{-\frac{b}{\lambda_k}} \end{aligned} \tag{4.3}$$

where $\mathcal{IG}(\cdot)$ is the inverse-gamma distribution, i.e. the distribution of the reciprocal of a gamma distributed stochastic variable. This means that $\frac{1}{\lambda_k}$ follows a gamma distribution. The hyperparameters a and b model the shape and scale of the prior for λ_k. Figure 4.3 shows some inverse-gamma distributions for several hyperparameter settings.

Following a maximum-a-posteriori approach, Tan *et al.* have derived the following cost function which should be minimized

$$C(\mathbf{W},\mathbf{H},\boldsymbol{\lambda}) = \frac{1}{\varphi}D_{\mathrm{KL}}(\mathbf{V}||\mathbf{WH}) + \sum_{k=1}^{K}\left(\frac{1}{\lambda_k}\left(\sum_{f=1}^{F}w_{fk} + \sum_{n=1}^{N}h_{kn} + b\right) + c\log\lambda_k\right) + \text{cst} \tag{4.4}$$

where $c = F + N + a + 1$ and $\boldsymbol{\lambda}$ is the vector containing all relevance parameters [9]. Note that both the l_1-norm of the objects $\mathbf{w_k}$ and the l_1-norm of the activations $\mathbf{h_k}$ are being penalized. As a result, large l_1-norms are only justified when a relevance parameter is large.

The parameter φ is a trade-off between accurate reconstruction, on the one hand, and of pruning irrelevant objects, on the other hand [9]. Two extreme cases illustrate the effect of φ:

- $\varphi \approx 0$ When φ is equal to or near zero, the term corresponding to the relevance weights will barely contribute to the cost function. As a result, only the Kullback-Leibler divergence $D_{\mathrm{KL}}(\mathbf{V}||\mathbf{WH})$ will be penalized. In other words, when the parameter φ approaches zero, regular nonnegative matrix factorization is performed and no irrelevant objects will be pruned away.
- $\varphi \approx \infty$ When φ is very large, only the term corresponding to the relevance weights will contribute to the cost function. The cost function has a global minimum when all elements of both $\mathbf{W}$ and $\mathbf{H}$ are set to zero. The model order will thus be driven to zero. Obviously, this will lead to an unfavourable solution since the reconstructed data does not resemble the original data $\mathbf{V}$ at all.

Tan *et al.* have developed an algorithm within this model which minimizes the cost function of equation (4.4) [9]. This algorithm can be found in algorithm 4 in appendix C.2. Here, the

repmat-operator denotes that a column or row should be replicated to obtain a matrix with the same dimensions as $\mathbf{W}$ or $\mathbf{H}$. Algorithm 4 is equivalent to the algorithm derived by Tan *et al.*, but computationally more efficient; less multiplications are necessary when a simple summation is performed and the replication operation is performed after element-wise division.

An up-close inspection of this algorithm leads to the following insights. Firstly, each relevance parameter affects exactly one column of $\mathbf{W}$ and one row of $\mathbf{H}$, as expected. When a relevance parameter λ_k of an object is small, the denominators for the k^{th} column of $\mathbf{W}$ and k^{th} row of $\mathbf{H}$ become large. As a result, the elements in $\mathbf{w_k}$ and $\mathbf{h_k}$ will be driven to zero. When the relevance parameter of an object is large, the second term in the denominator will be negligible and regular KL-NMF is executed (see algorithm 2 in appendix C.1). Secondly, the relevance parameters are dependent on the l_1-norm of the objects $\mathbf{w_k}$ and the corresponding activations $\mathbf{h_k}$. Whenever a column $\mathbf{w_k}$ and the corresponding row $\mathbf{h_k}$ have relatively large elements, it is assumed that the corresponding component $\mathbf{C_k}$ is relatively important in the decomposition, which is why a large relevance parameter λ_k should be assigned to it and vice versa. Thirdly, as explained earlier, when a component $\mathbf{C}_k$ is irrelevant, the elements of the corresponding object $\mathbf{w_k}$ and activations $\mathbf{h_k}$ are driven to zero. However, the corresponding relevance weight λ_k is not driven to zero, but to a *lower bound* which can be derived from the update rule for λ_k [9]. A component is most irrelevant when all its corresponding elements w_{fk} and h_{kn} are equal to zero. When this happens, the corresponding relevance weight will be set to the following lower bound

$$\lambda_{\text{LB}} = \frac{b}{c} = \frac{b}{F + N + a + 1}. \tag{4.5}$$

To speed up the algorithm, $\mathbf{w_k}$ and $\mathbf{h_k}$ can be disregarded when the corresponding relevance weight λ_k is equal to or sufficiently near the lower bound defined in eq. (4.5). This dramatically reduces the dimensions of $\mathbf{W}$ and $\mathbf{H}$ and speeds up the computation significantly. Tan *et al.* have not mentioned this optimization.

Results

Two different experiments have been conducted to illustrate the modus operandi of automatic relevance determination .

In the first experiment, automatic relevance determination has been applied to a single-speaker speech segment of 100 seconds for 5000 iterations. The spectrogram of this segment has 512 frequency bins and 2254 times frames after voice activity detection. When factorizing spectrograms with only one speaker present, it is sufficient to set K_{init} to a value of 50. In other words, a speaker is assumed to be fully characterized by at most 50 objects $\mathbf{w_k}$. The values for the hyperparameters a and b are equal for all components and have been set to 100 and 6.25 respectively. The value for hyperparameter b has been computed using the method of moments as described by Tan *et al.* [54]. In figure 4.4, the progression of the relevance weights is presented. As can be seen, several relevance weights are gradually driven to the lower bound from equation (4.5). In this case, the lower bound is equal to

$$\lambda_{LB} = \frac{6.25}{512 + 2254 + 100 + 1} = 0.00217. \tag{4.6}$$

The remaining relevance weights stagnate and correspond to relevant components. In this case, the inherent model order K^* is estimated to be 12. Notice that although a few relevance

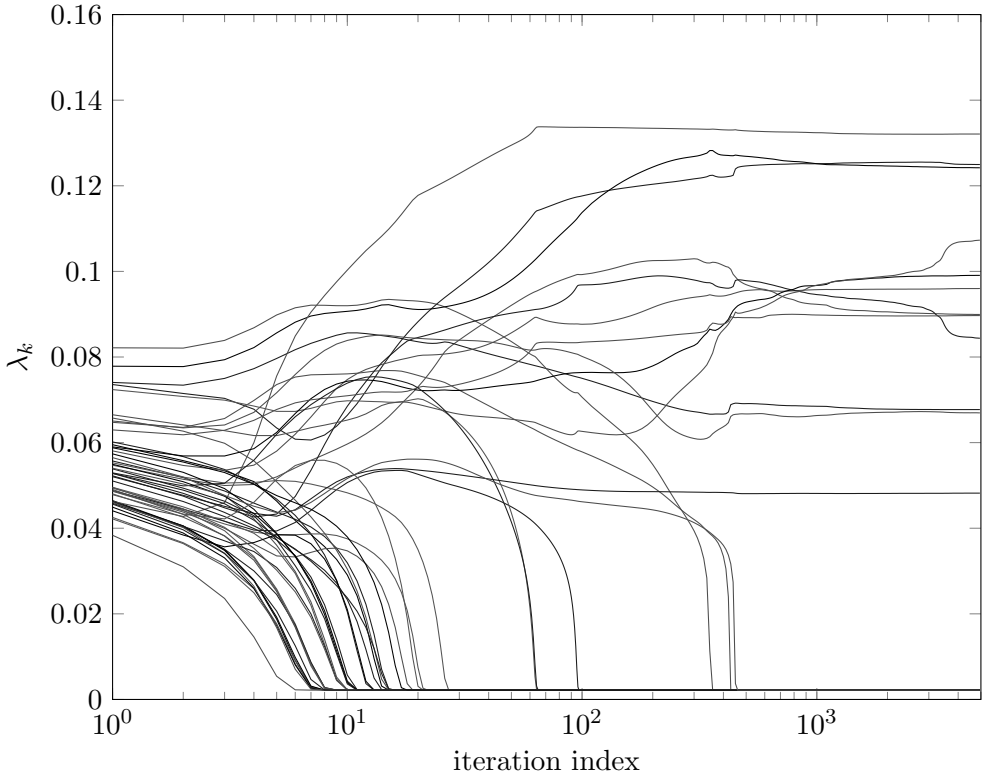


Figure 4.4: Evolution of the relevance weights λ_k for 5000 iterations on a log-linear scale for ARD on a spectrogram from a speech segment of 100 seconds. Parameters: $K_{init} = 50$, $\varphi = 0.5$, $a = 100$, $b = 1$

	10s				100s				1000s			
S_{ID}	1	2	3	4	1	2	3	4	1	2	3	4
Speech	229	250	210	256	2254	2362	2495	2451	23194	23544	25208	24052
Silence	84	63	103	57	871	763	630	674	8056	7706	6042	7198
Total	313	313	313	313	3125	3125	3125	3125	31250	31250	31250	31250

Table 4.1: VAD results for each speech signal in the ARD experiment. 'Speech' indicates the number of frames that were classified as containing speech and 'Silence' indicates the number of frames that were classified as not containing any speech.

parameters λ_k still change slightly, the final estimation for the model order is already found after 500 iterations.

In the second experiment, the influence of the length of training data and the value of φ are examined. In order to do so, the model order has been estimated for four different speaker and three different signal lengths; 10 seconds, 100 seconds and 1000 seconds. The algorithm was run for four different values of φ; 0.1, 0.5, 1 and 2. The algorithm has been executed 10 times for each set of parameter values. Prior to automatic relevance determination, voice activity detection was performed. Table 4.1 indicates how many frames were classified as speech and noise by voice activity detection for each signal.

The results can be found in figure 4.5. The following trends can be observed.

- As expected, the estimated model order tends to decrease when φ increases. This comes as no surprise since the parameter φ determines at what relevance level an object should be pruned away. Estimated model orders fall in the range of 10 to 20 when φ is set to 0.5. We have found that this is a good compromise between feature dimensionality and performance.
- The model orders estimated for signal lengths of 1000 seconds and 100 seconds do not differ significantly, which is a good sign. However, the estimated model orders for a signal of 10 seconds is about half as large. This is probably because not all sounds are contained in a signal of 10 seconds. As a result, at least 100 seconds of audio data should be used to estimate the model order. If so, the model order can be estimated fairly accurate.

Finally, we'd like to mention that the signal amplitude affects model order estimation as well. The estimated inherent model order K^* is different when ARD is performed on signals that are identical, but for a constant scale factor. This is because more components are deemed to be relevant if the elements of the data $\mathbf{V}$ are larger due to the definition of the relevance parameters. To account for this, $\mathbf{V}$ needs to be normalized with respect to the average frame energy.

Final notes

Once the right parameters have been found, automatic relevance determination is a robust way of discovering the inherent model order K^* as long as a sufficient amount of speaker data ($> 100s$) is available. The obtained inherent model order K^* is subsequently used for factorization of $\mathbf{V}$ with Bayesian inference as explained in section 3.3. However, performing order estimation and factorization sequentially as done in this thesis is suboptimal since a factorization of $\mathbf{V}$ is found during both ARD and BNMF. A better solution consists combining ARD and BNMF in a single algorithm by introducing relevance parameters in the derivation of BNMF. Such a derivation is beyond the scope of this thesis, and we leave it for future work.

4.4 Feature Extraction: Group Sparsity nonnegative matrix factorization

As already briefly mentioned in section 4.2.1, features need to be extracted from speech segments in the learning phase as well as the identification phase. On the one hand, features are necessary during the learning phase for training classifiers. On the other hand, these trained classifiers need to categorize features that are extracted from speech segments during the identification phase. In this section, the step-by-step process of extracting features from a spectrogram $\mathbf{V}_{\mathbf{VAD}}$, which are derived from a set of speaker dictionaries of objects that have been acquired through NMF, is explained.

In figure 4.6, a more detailed depiction of the feature extraction process is shown. The set of necessary speaker dictionaries $\mathbf{W} = \{\mathbf{W_1}, ..., \mathbf{W_S}\}$ are stored in advance, since they are necessary during feature extraction. Once these dictionaries are available, features can be extracted from a spectrogram $\mathbf{V}_{\mathbf{VAD}}$.

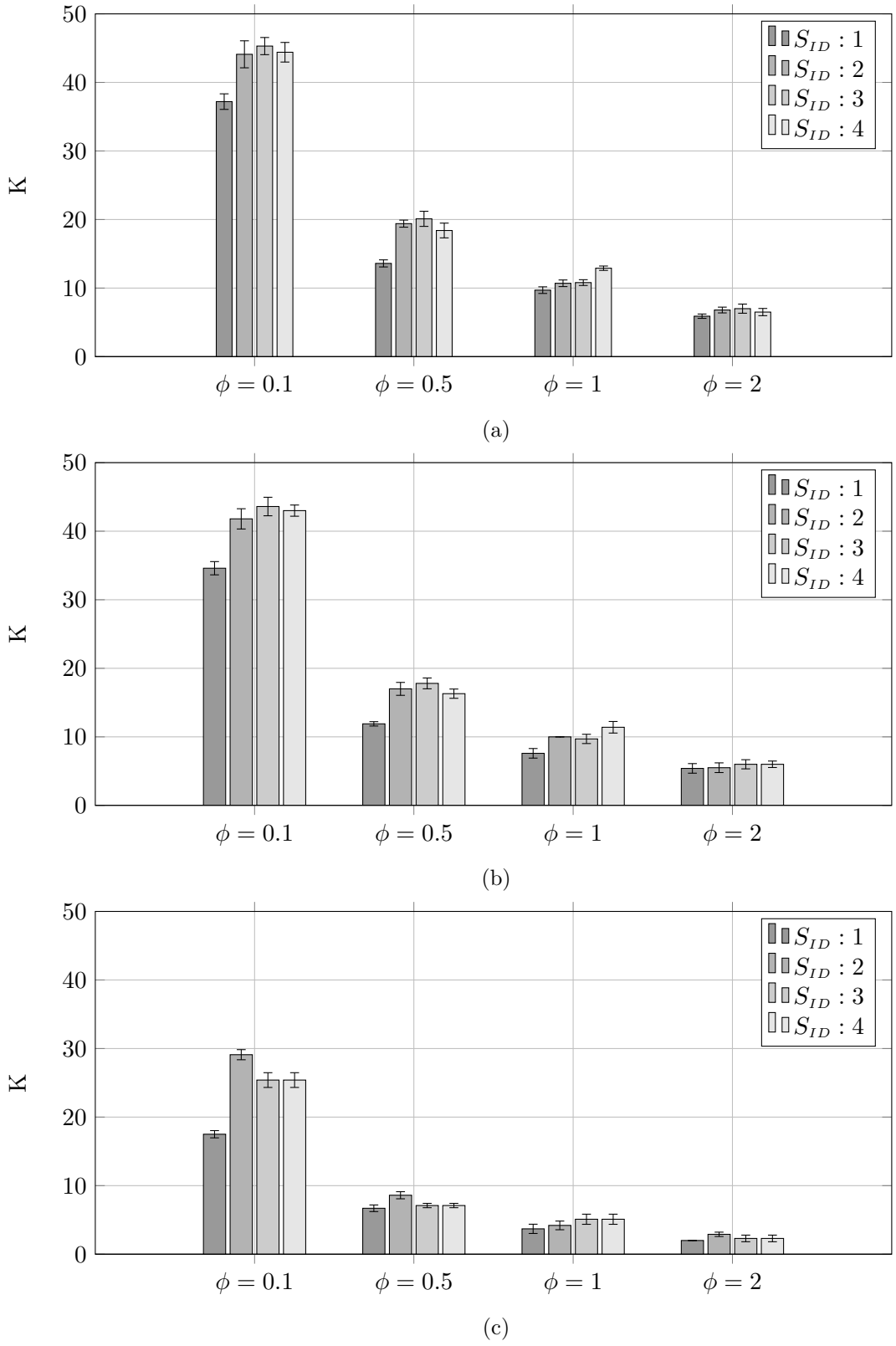


Figure 4.5: ARD results for 4 different speakers, 4 different values for φ (0.1, 0.5, 1 & 2) and three different signal lengths; (a)1000s, (b) 100s & (c) 10s. The estimated model order is averaged over 10 runs. Error bars indicate the standard deviation.

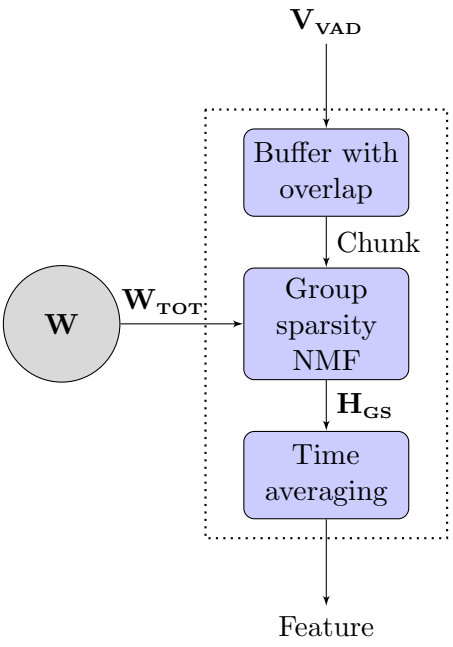


Figure 4.6: Detail of feature extraction with GS-NMF

Buffer

Before any features can be extracted, a **buffer with overlap** is applied. This buffer partitions incoming frames into sets of J successive frames. In this section, the term block denotes the those sets. An overlap of $I-1$ is chosen so that each block is shifted exactly one frame relative to its preceding block. The feature that will be extracted from this block is going to be assigned to the $(\frac{J+1}{2})^{th}$ frame of the block. This is the central frame of the block. It is therefore that J should be an odd number. Otherwise, each feature would need to be distributed over two consecutive frames.

The choice of J is important. For a particular frame, it determines the amount of frames that will affect the feature corresponding to that frame. It is a trade-off between temporal accuracy and smoothness. When J is large, brief audio events will not be identified. However, features will show less variability since many frames contribute to a feature. On the contrary, when J is small, short speech segments might be correctly identified, but features will be more prone to noise. To illustrate this, imagine a speech signal where a speaker is speaking and simultaneously, a door is slammed shut. When J is large, the effect of the door will not be as dramatic as when J would be small. As J increases, however, the computational complexity increases as well. In this thesis, J is set to 15. Such a block size corresponds to 7680 samples or 480 milliseconds when the STFT window length N_{FFT} is equal to 1024 samples with an overlap of 512 samples.

One can note as well that J determines the amount of delay as well when speaker identification is used real-time. Suppose that any following computations regarding feature extraction and classification are negligible. A speaker identity estimate can only be produced when enough frames have passed to fit in a block. Since such an estimate corresponds to the center frame of a block, these estimates will have a delay of at least $\frac{J.N_{FFT}}{2.f_s}$.

Group Sparsity Nonnegative Matrix Factorization

Whenever enough frames have been buffered into a block, a feature can be extracted from it. In order to do so, a technique called **group sparsity nonnegative matrix factorization (GS-NMF)** is applied. Hurmalainen *et al.* have developed this technique for use in speech recognition [6].

The goal of GS-NMF is similar to ordinary NMF; given some data $\mathbf{V}$ and a dictionary $\mathbf{W}$, which has been learnt in advance, to find the time activation matrix $\mathbf{H}$ which minimizes the Kullback-Leibler divergence between $\mathbf{V}$ and $\mathbf{WH}$. However, in GS-NMF, groups of basis vectors can be defined on the known dictionary $\mathbf{W}$ and the objective is to find a decomposition where the number of active groups within a block is limited, hence the term group sparsity [6]. In addition to group sparsity, regular sparsity is enforced as well, i.e. the number of active objects within a frame is limited as well [6]. In summary, the obtained activation matrix $\mathbf{H_{GS}}$ has a limited number of active groups within a block and a limited number of active objects within each frame.

GS-NMF is ideal for speaker identification in a typical conversation. Occasionally, some impolite speakers might speak simultaneously, but generally speakers will take turns in speaking. In that case, only one speaker will be active at any point in time. If the combined dictionary is defined as the concatenation of all individual speaker dictionaries and groups are defined along the boundaries of these speaker dictionaries, it makes sense to find a decomposition in the previously stated assumption where only one group is active. Thus, the use of group sparsity is justified. When speakers do speak simultaneously, we shall see that GS-NMF can still be used if combined with source separation (see chapter 5).

The dimensions of the combined dictionary

$$\mathbf{W_{TOT}} = \begin{bmatrix} \mathbf{W_1} & | & \mathbf{W_2} & | & \dots & | & \mathbf{W_{S-1}} & | & \mathbf{W_S} \end{bmatrix}. \tag{4.7}$$

are $F \times K_{TOT}$, where K_{TOT} is the total amount of objects in all dictionaries. The **group allocation matrix** $\mathbf{G_B}$, which is a boolean matrix of dimension $G \times K_{TOT}$, contains the definition of the G groups. In this matrix, 1's indicate that an object belongs to a certain group and 0's indicate that an object belongs to another group. For example, the group allocation matrix shown in equation (4.8) implies that a total of 9 objects are divided in three groups; the first three objects belong to group G_1, the four subsequent objects belong to group G_2 and the two final objects belong to group G_3.

$$\mathbf{G_B} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{4.8}$$

The cost function of GS-NMF, shown in (4.9), which has been proposed by Hurmalainen *et al.* contains the Kullback-Leibler divergence and two additional terms; a term c_{gs}, enforcing group sparsity, and a term c_1, enforcing regular sparsity [6].

$$\begin{aligned} C_{GS} &= D_{KL}(\mathbf{V_{VAD}} \,||\, \mathbf{W_{TOT}H_{GS}}) + c_{gs} + c_s \\ c_{gs} &= \lambda_g \left\| \sqrt{\mathbf{G_B H_{GS}}^{.2}} \right\|_1 \\ c_s &= \|\mathbf{\Lambda_1} \cdot \mathbf{H_{GS}}\|_1 \end{aligned} \tag{4.9}$$

The first term c_{gs} enforces group sparsity. It measures the l_2-norm of the activations within groups per frame and sums these over time and groups [6]. To illustrate the effect of c_{gs}, assume that the activations shown in eq. (4.10a) have been found using regular NMF and that groups are defined as in eq. (4.8). Since there are 9 objects, $\mathbf{H}$ has dimensions $9 \times J$. J is equal to 7. Dashed lines indicate the groups in $\mathbf{H}$. The penalty c_{gs} for this activation matrix can be found in equation (4.10b).

$$\mathbf{H} = \left[\begin{array}{ccccccc} 0.9 & 0.8 & 0 & 0 & 0 & 0.9 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0.2 & 0.9 & 0 & 0 \\ \hdashline 0.9 & 0 & 0.1 & 0 & 0.1 & 0 & 0 \\ 0 & 0.4 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.4 & 0.5 & 0 & 0.1 \\ 0 & 0.1 & 0 & 0.1 & 0.1 & 0 & 0 \\ \hdashline 0 & 0 & 0 & 0 & 0.3 & 0.3 & 0 \\ 0 & 0.2 & 0 & 0.2 & 0 & 0 & 0.1 \end{array}\right] \tag{4.10a}$$

$$c_{gs} = \lambda_g \left\| \begin{array}{ccccccc} 0.9 & 0.8 & 0.71 & 0.2 & 0.9 & 0.9 & 0 \\ 0.9 & 0.41 & 0.1 & 0.41 & 0.52 & 0.1 & 0.1 \\ 0 & 0.2 & 0 & 0.2 & 0.3 & 0.3 & 0.1 \end{array} \right\|_1 = \lambda_g \left\| \begin{array}{c} 4.41 \\ 2.54 \\ 1.1 \end{array} \right\|_1 = 8.05\lambda_g \tag{4.10b}$$

Some remarks can be made here. First, the high cost associated with the first frame stands out, it accounts for almost a fourth of the total cost c_{gs}. This high cost is the result of groups G_1 and G_2 which are simultaneously active. If the objects with activations equal to 0.9 would belong to the same group instead, the group sparsity cost associated with the first frame would be reduced from $1.8\lambda_g$ to $1.27\lambda_g$. It is clear that when the cost function is minimized and λ_g is sufficiently high, solutions with large group sparsity are favored. Second, note that because the l_2-norm is being used to evaluate the group norm within frames, regular sparsity is relatively insignificant for the cost c_{gs}. For example, if the l_1-norm would have been used, the cost associated with group G_1 and the third frame would have been 0.8 instead of 0.71. Finally, note that group sparsity is assessed on a frame basis. Hurmalainen *et al.* propose to sum the activations over all frames and using this $K_{TOT} \times 1$ vector $\mathbf{H}_\Sigma$ instead of the full $K_{TOT} \times N$ activation matrix $\mathbf{H_{GS}}$ within the cost c_{gs} [6]. In that case, group sparsity is assessed on an entire block. However, when multiple speakers are present in a single block of frames, block-based GS-NMF might lead to ambiguous results. In this case, frame-based GS-NMF should be used where $\mathbf{H_{GS}}$ is not summed over frames.

The second additional term c_s of the cost function shown in eq. (4.9) enforces regular sparsity. This term ensures that within each frame but a limited number of objects are active. The matrix $\mathbf{\Lambda}_1$ contains the relative contribution of the sparsity cost c_s for each activation. Even though it is possible to choose a different sparsity cost contribution for each individual activation h_{kn}, we set each element of Λ_1 to a fixed parameter λ_1.

The algorithm for block-based GS-NMF derived by Hurmalainen *et al.* for the cost function shown in eq. (4.9) and the frame-based variant of GS-NMF are shown in algorithms 5 and 6 in appendix C.2 [6].

Time averaging

The obtained activation matrix $\mathbf{H_{GS}}$ for a block of frames has dimensions $K_{TOT} \times J$. The activations are averaged over time to obtain a feature vector with K_{TOT} features. The dominant group in this feature vector is assumed to be a correct indication of the speaker identity. However, other objects belonging to other speakers may have large activations as well because speaker objects are not fully independent of each other. However, we should not pay too much attention to these dependencies. They are accounted for in the classifier.

Results

Figure 4.7 shows features extracted for a $J = 15$ or 0.48 seconds for regular NMF, frame-based GS-NMF and block-based GS-NMF. The features were extracted from a segment of 4.25 seconds containing two utterances from two different speakers. The combined speaker dictionary was learnt prior to this experiment and contains 20 objects; the first 10 objects belong to the first speaker and the last 10 objects belong to the second speaker. The group sparsity parameter has been set to 10, the regular sparsity parameter has been set to 0.1.

Block-based and frame-based GS-NMF lead to similar features. This is probably because there is no part where two speakers speak occur during the same block. However, it is clear that both GS-NMF algorithms are able to discover the correct speaker identity.

4.5 Classification: Support Vector Machines

Section 4.4 explains how speaker-dependent features can be extracted from a segment of a speech spectrogram. A classifier is needed to determine the target attribute—in this case the speaker identity—from the other attributes or features contained in the D-dimensional feature vector $\mathbf{x}$. A discriminative classifier such as a **support vector machine** does this task by dividing the entire feature space in two regions corresponding to two different classes. The goal of the classifier $f(\cdot)$ is to output the target attribute y, which will henceforth be called the label, as shown in eq. (4.11).

$$y = \text{sign}\left(f(\mathbf{x})\right) \tag{4.11}$$

To train a classifier, a large number of features and their corresponding labels are needed. In this chapter, the process of classification is explained. In this work, we have chosen support vector machines [23] for their ease-of-use and low computational cost when the feature space is high-dimensional [55].

Support vector machine

In section 2.5, support vector machines have already been introduced. They are widely used in various machine learning problems. To recapitulate, a support vector machine is a binary classifier which defines a hyperplane in the feature space. This hyperplane represents the boundary between two classes. Since the hyperplane is a plane, features belonging to different classes should be linearly separable. This is a drastic requirement. However, a trick can be applied to circumvent this requirement. A non-linear transformation $\phi(\mathbf{x})$ can be applied to map the feature space to a new space so that features are linearly separable in this transformed space. It is possible for this transformed space to have a higher dimensionality. The non-linear

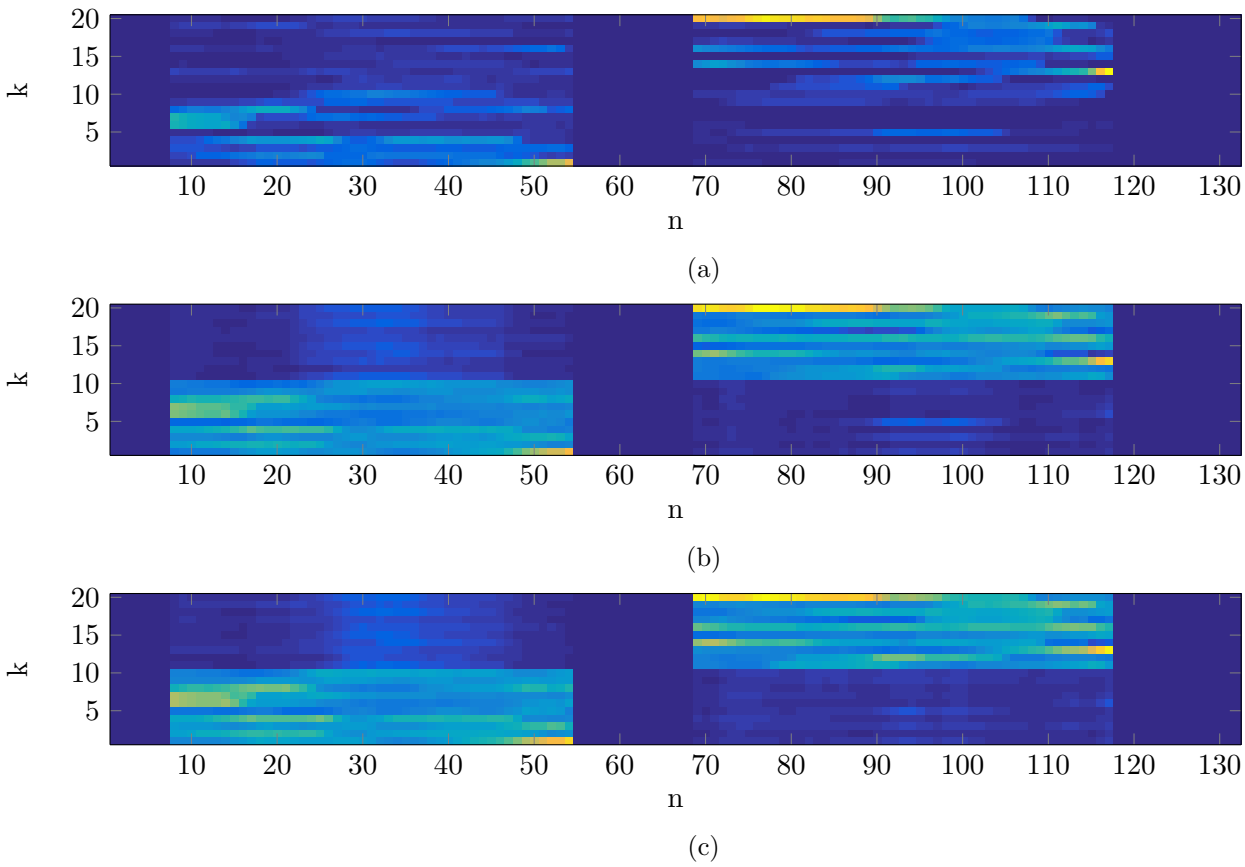


Figure 4.7: Features extracted from a speech segment of 4.25 seconds obtained with (a) regular NMF, (b) Frame-based GS-NMF and (c) Block-based GS-NMF

transformation can be stated as follows

$$\mathbf{z} = \phi(\mathbf{x}) \tag{4.12}$$

where $\mathbf{z}$ represents transformation of feature $\mathbf{x}$.

However, the computation of the transformation of features can be computationally expensive. Actually, it isn't even necessary to directly compute the transformation $\phi(\mathbf{x})$. As mentioned in section 2.5, scoring consists of a simple dot-product which is fast to compute [24]. Without non-linear transformation, the dot product can be stated as follows

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \tag{4.13a}$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i. \tag{4.13b}$$

where $\mathbf{x}_i$ are the support vectors, y_i are their respective class (either -1 or 1). The vector $\mathbf{w}$ is the normal vector to the bounding hyperplane and b is a parameter which defines the offset $\frac{b}{\|\mathbf{w}\|}$ of the hyperplane with respect to the origin [24]. The discriminant function $f(\mathbf{x})$ evaluates if the feature $\mathbf{x}$ lies on the same side as where the normal vector $\mathbf{w}$ points—if $f(\mathbf{x})$ is positive—or on the opposite side—when $f(\mathbf{x})$ is negative. Without going into further detail, the normal

vector $\mathbf{w}$ is a linear combination of a limited number of features called the support vectors $\mathbf{x}_i$ as can be seen in equation (4.13b). The support vectors are the features which lie closest to the separating hyperplane. When the non-linear transformation is applied, the discriminant function can be expressed by [24]

$$\begin{aligned} f(\mathbf{x}) &= \sum_i \alpha_i y_i \phi(\mathbf{x}_i)\phi(\mathbf{x}) + b \\ &= \sum_i \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b. \end{aligned} \tag{4.14}$$

This expression states that if a kernel function $k(\mathbf{x}_i, \mathbf{x})$ can be found, a support vector machine can be evaluated in a transformed space without having to directly compute the transformation of the features $\phi(\mathbf{x})$. As such, non-linear transformation of the feature space can be done efficiently with the kernel trick.

Various types of kernels exist, but all support vector machines in this thesis apply a polynomial kernel as shown in eq. (4.15). The kernel determines which bounding surfaces are feasible. The polynomial kernel is defined as

$$k(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1^T \mathbf{x}_2 + c)^d \tag{4.15}$$

where c is a constant parameter and d is the degree of the polynomial kernel. The computational complexity is larger for high-degree polynomial kernels but the flexibility of the bounding surface increases as well.

Multiclass SVM

Since support vector machines use a separating bounding surface defined by $\mathbf{w}$, they are naturally binary classifiers. This poses a problem as we need to differentiate between more than two classes in speaker recognition. Several extensions which allow multiclass support vector machine classification exist [], yet the most straightforward method is the **one-versus-all** technique [5]. Instead of training a single support vector machine, C different support vector machines are trained, i.e. one for each class. Each of these support vector machines is trained using the same set of feature, namely the set containing all features from every possible class. However, the labelling for the training of each support vector machine is different; each feature which belongs to the class whose support vector machine is being trained is given a positive label $\{+\}$ and all other features are given a negative label $\{-\}$. Training of support vector machines is done using the fast matlab function `fitcsvm` [56]. All the support vector machines are stored so that they can be used later on for predicting labels.

The prediction of a label of a feature happens as follows; each of the support vector machines —one for each class—is applied with the matlab function `predict` [57]. Each support vector machine predicts whether a feature belongs to the class corresponding to the support vector machine or not. In the ideal case, only one support vector machine indicates that the feature is classified as positive. In that case, the prediction of the feature's label is straightforward; it is the label corresponding to the only support vector machine with a positive output.

[5] The one-versus-one technique is an alternative multiclass classification solution for support vector machines. A total of $\frac{C(C-1)}{2}$ classifiers are trained. Each of these classifiers trains a bounding surface between feature clouds from exactly two classes. This technique may be applied instead of the one-versus-all technique.

However, multiple support vector machines can output positive labels, indicating that the feature belongs to more than one class. In the assumptions which can be found in section 4.1, we have stated that speakers do not speak simultaneously. In this scenario, a set of support vector machine outputs with multiple positive labels is clearly a fault. In that case, an additional certainty measure is needed to choose the best fitting class. The second output of the function `predict` is the value of the classifier function $f_c(\mathbf{x})$. When multiple identities are possible, the best identity estimate can be found by maximizing $f(\mathbf{x})$ over all classes with a positive label

$$C^* = \arg\max_c f_c(\mathbf{x}) \quad , \forall c : y_c = \{+\} \tag{4.16}$$

where C^* is the final class estimate, $f_c(\mathbf{x})$ is the classifier score for feature $\mathbf{x}$ and c can be any class with a positive label y_c.

Additionally, the classifier score can be used as well for verification. It is possible to fit a score-to-posterior transform functionin Matlab [57]. As such, the posterior probabilities of a correct classification are available. If this posterior does not exceed a certainty threshold, a label $\varnothing$ can be assigned to the feature indicating that no known speaker is recognized.

4.6 Experiments and results

In this section, the techniques for monaural speaker identification that have been put forth in this chapter are assessed. Speech data is used from two data sets; the CHiME data set and the panel meeting data set. More information about these data sets can be found in appendix A.

Experiment 1: Speaker identification on CHiME database

As a first experiment, we evaluate the object-based speaker recognition on simulated conversations composed of utterances from the CHiME data set. The most important parameter is the set size of known speakers S. The algorithm has been analysed for set sizes in the range of 2 to 14 known speakers. The size of the data set U is also very important. As we've mentioned in appendix A.1, each speaker has a total of 500 utterances. However, to speed up the computations of this experiment, only 200 utterances were used. For each speakers, these 200 utterances were divided into ten sets of 20 utterances. Ten-fold crossvalidation has been performed on these sets: speaker identification has been evaluated ten times and for each fold, nine of the ten subsets are used for training and the remaining subset is used for validation. This means that 180 utterances have been used per speaker to construct a speaker model. Since each utterance lasts about 1.5 seconds, a speaker model is trained on about 4 minutes and 30 seconds of speech. The 20 remaining utterances per speaker are used to simulate a conversation; all utterances from all speakers are concatenated in a random order with an intermittent silence of 0.5 seconds. Validation could also have been performe utterances drawn from the set of 300 unused utterances, but we have chosen to follow the standard procedure for crossvalidation. All speakers in the set are *male*. This is important because speaker identification between men and women is generally an easier task than identification between same-gender speakers. Other parameter values can be found in table 4.2.

Experiment 2: Speaker identification on Panel Meeting database

The conversations used in experiment 1 are not actual conversations but rather a concatenation of individual of utterances. Also, the signals from the CHiME database have been recorded

STFT	N_{FFT}	1024
	overlap	50%
ARD	K_{init}	30
	φ	0.5
	N_{iter}	2000
	a	100
	b	6.25
BNMF	a	1
	b	1
GS-NMF	J	15 (= 480ms)
	N_{iter}	20

Table 4.2: Parameter settings for Experiments 1 and 2

in a sound-proof studio. As a result, these signals contain almost no noise and are not very representative for real-live recordings. To test the robustness of the algorithm, we have evaluated the performance with real-live recordings from the Panel Meeting database (see appendix A.2). The parameters in this experiment are the same as for experiment 1 and can be found in table 4.2. However, the set size of known speakers and the amount of data per speaker is different. The maximum set size of known speakers is eight since not enough data was available for the last two speakers (see table A.2 in appendix A.2). For each speaker, 6 minutes and 30 seconds of recordings have been used for training speaker models and 1 minute and 30 seconds of recordings have been used for validation. Although the set size is larger than for experiment 1, we estimate that the true amount of speech is comparable since the CHiME database contains read speech and the Panel Meeting database contains spoken speech. Read speech generally contains proportionally less silence than spoken speech.

Results

The results for both experiment can be found in figures 4.8 and 4.9. As expected, the results for the clean signals of the CHiME database are significantly greater than those for the Panel meeting database. The Panel meeting database is far more noisy. In experiment 1, we have excellent performance for set sizes smaller or equal to 10. For larger set sizes, there is a performance degradation. We believe that the drop in performance can be explained by the large dimensionality of the features for larger sets of speakers.

As a comparison, we add the results of object-based speaker recognition presented by Joder *et al.* [7]. This technique, although object-based, differs from ours. Joder *et al.* propose to factorize a signal containing many speakers during training. The resulting dictionary contains objects which are characteristic for many speakers. During identification, the activations $\mathbf{H}$ are sought for the given dictionary extracted during training. These activations are the frame-wise features called H-NMF [7]. They also propose a different set of features which consists of MFCC extraction from the reconstructed frame $\hat{\mathbf{v}}_\mathbf{n} = \mathbf{W}\mathbf{h}_\mathbf{n}$ [7]. These features are called MFCC-NMF. Joder *et al.* also perform a majority voting over a context window of 1 second [7]. A comparison between the results from Joder *et al.* and our results for the CHiME database can be found in table 4.3 [7]. The set of known speakers contains 8 speakers. The data used by Joder *et al.*

Joder *et al.* [7]	H-NMF (1s Vote)	97.27%
	MFCC-NMF (1s Vote)	98.63%
Our results (CHiME)	GS-NMF ($I = 15 = 0.48$s)	98.31%

Table 4.3: Comparison of speaker identification success rate for a speaker set of 8 speakers.

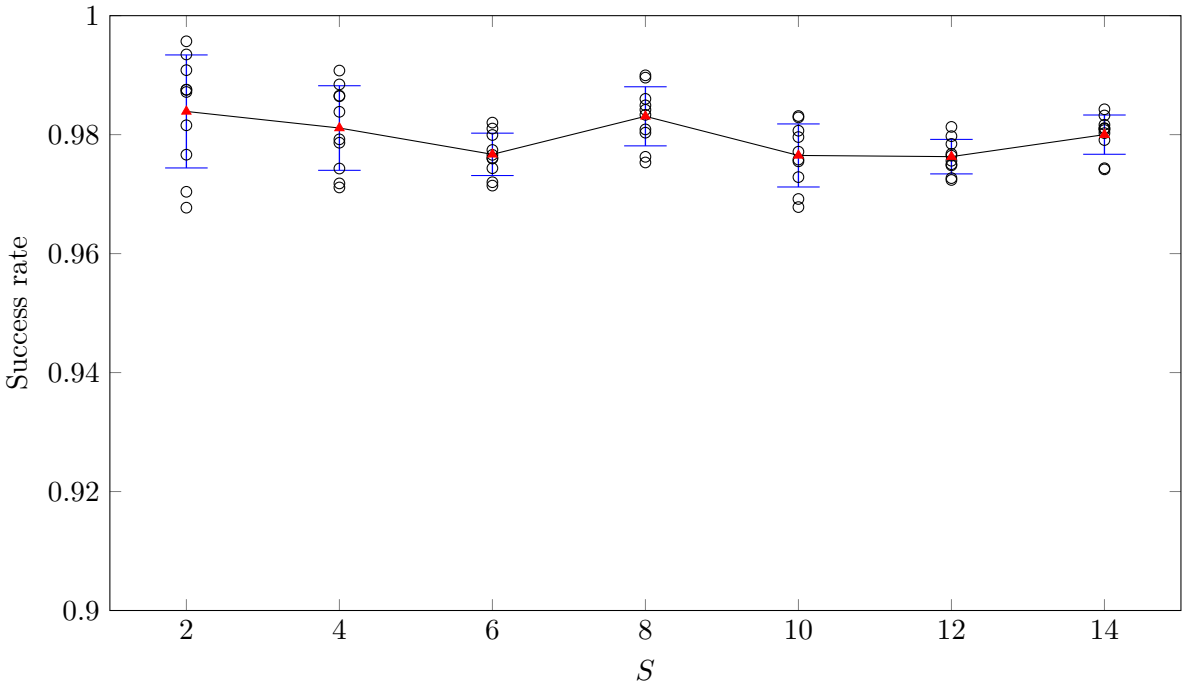


Figure 4.8: Performance results for 10-fold crossvalidation speaker identification on the CHiME database. Black circles denote the success rate for individual folds, blue error bars indicate the standard deviation and red triangles indicate the average success rate.

originates from the TIMIT database which has a similar quality as the CHiME database.

The following remarks can be made. The results are very similar. However, Joder *et al.* perform majority voting over a context window of 1s. We do not perform any voting, but a feature is extracted from a block of J frames. In our experiment, J was set to 15 frames which corresponds to 0.48 seconds. So we see that our system gives a similar performance for a shorter context window. Also, our experiments were performed on speaker sets which contain only male speakers. Joder *et al.* have performed their experiments on a speaker set containing four male speakers and four female speakers. To sum up, we have achieved a similar performance result as Joder *et al.*, but for a harder task. However, disadvantages of our speaker identification system are the increased computational complexity due to model order estimation and Bayesian inference, on the one hand, and the dimensionality increase for increasing speaker set sizes, on the other hand. The main advantage of both our technique and the technique from Joder *et al.* is that no universal background model is needed and speaker models can be derived using only about 5 minutes of speech recordings per speaker.

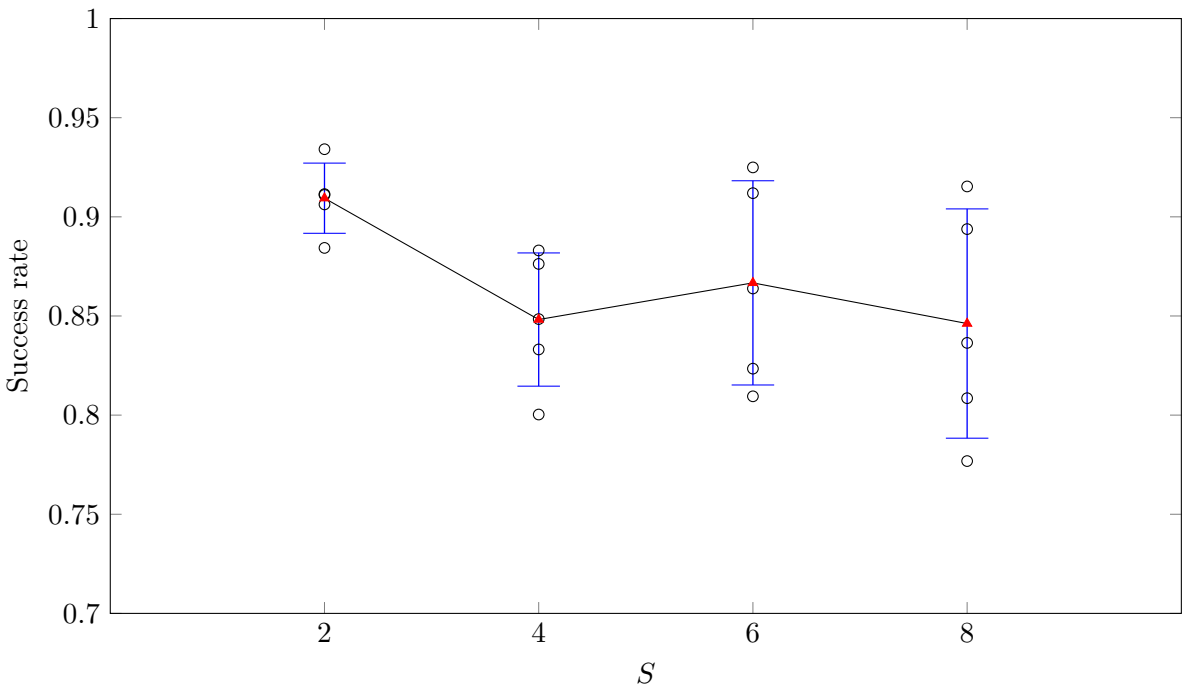


Figure 4.9: Performance results for 5-fold crossvalidation speaker identification on the Panel Meeting database. Black circles denote the success rate for individual folds, blue error bars indicate the standard deviation and red triangles indicate the average success rate.

4.7 Conclusion

In this chapter, we have shown how speaker dictionaries $\mathbf{W_s}$ can be used for speaker identification. We proposed solutions for key issues; automatic relevance determination, feature extraction and multiclass classification with support vector machines.

Firstly, we have shown how the correct inherent model order K^* can be found for a speaker given a spectrogram $\mathbf{V_s}$ using a technique developed by Tan *et al.* [9]. When performing automatic order determination, it is important to have at least 100 seconds of actual speech recordings per speaker, to normalize to average frame energy and to set the parameter φ to an appropriate value.

Secondly, we have shown how features can be extracted based on speaker dictionaries $\mathbf{W_s}$ with group sparsity nonnegative matrix factorization proposed by Hurmalainen *et al.* [6]. The important parameters here are the size of the block of frames J, the importance of group sparsity λ_g and the importance of regular sparsity λ_1.

We have shown that our technique produces competitive results, especially for speaker sets below 10 speakers, compared to the object-based speaker identification system from Joder *et al.* [7]. However, our proposed speaker identification method only works when speakers do not speak simultaneously. Evidently, such an assumption is unrealistic if speaker identification is applied using real world systems. In the next chapter, we will see how our proposed technique can be used when speakers do speak simultaneously.

Chapter 5

Joint speaker localization, enhancement and identification

5.1 Introduction

In chapter 4, an object-based approach to speaker identification has been proposed. We have stated that such a method is only applicable if speakers do not occur simultaneously. However, in this chapter, we will show that our technique can be used in situations where speakers do speak concurrently if stereo audio signals are available. Building on the works of Mirzaei *et al.* and Ozerov *et al.*, we propose a modular system where speaker localization, source separation and speaker identification are performed sequentially [2, 1]. Since identification is performed on separated signals, the condition that identification should be performed on single-speaker segments is satisfied as long as source separation is successful.

The first module, speaker localization, is part of the initialization scheme proposed by Mirzaei *et al.* [2]. As explained in section 3.4, the number of speakers and the bearing angles for each of these detected speakers are obtained from the GCC-PHAT metric. These bearing angles are subsequently used to obtain initial estimates of speaker spectrograms $\mathbf{V_s^{BM}}$ through binary masking. Factorizations $\mathbf{W_s H_s}$ of these initial estimates, combined with an estimate of the mixing matrix $\mathbf{A}$, serve as an improved initialization for the parameters of the EM algorithm for source separation developed by Ozerov *et al.* [2]. Since these techniques have been explained in section 3.4, we will not go too much into detail regarding these techniques. However, unlike the original localization method proposed by Mirzaei *et al.*, we do propose a slight change so bearing angle estimates can be obtained on a per frame basis. Albeit less robust, we believe that per frame estimations are better suited for future real-time applications. The noisy estimates are removed using spatiotemporal segmentation; high-level assumptions such as minimum angular separation and maximum intersegmental silence are used to group spatiotemporal sound source detections into single-speaker segments. The second module separates each of the detected segments using the object-based EM source separation algorithm developed by Ozerov *et al.* that has been explained in chapter 3. The third module applies object-based speaker identification where the speaker dictionaries have been trained in advance using NMF with Bayesian inference and features are extracted through GS-NMF with the combined dictionary $\mathbf{W_{TOT}}$ from all known speakers. For more information about the works of Ozerov *et al.* and Mirzaei *et al.*, we refer the reader to chapter 3. For more information about object-based speaker identification, we refer the reader to chapter 4.

The remainder of this chapter is organized as follows. Section 5.2 contains an overview of the full joint system and how each of the modules operates. In section 5.3, we evaluate the performance of joint speaker localization, enhancement and identification on non-reverberated and reverberated simulated scenes for different sets of parameter values. In section 5.4, some conclusions are given.

5.2 Joint system schematic

Figure 5.1 shows a schematic of the different steps in joint speaker localization, enhancement and identification. Each of these steps are enclosed by a dotted line. An additional important step is the segmentation, which happens after localization and before enhancement. In this section, the full process is roughly explained. Binary masking and object-based source separation are applied as explained in chapter 3. However, in the joint system, localization and subsequent segmentation are slightly altered. These techniques are given additional attention below.

5.2.1 Auditive input

The microphones are placed in a room and record any sound produced in this room. The symbols $v_1^c(t)$ and $v_2^c(t)$ denote the left and right microphone signal respectively. Notice that these signals are the mixture signals. The spectrograms of these two signals, denoted by the symbols $\mathbf{V_1^c}$ and $\mathbf{V_2^c}$, are computed using STFT with a window length N_{FFT} of 1024 samples and an overlap of 50%. The set of both channel mixture spectrograms is denoted by $\mathbf{V^c}$. Any ensuing computations are done in the frequency domain.

5.2.2 Localization

Localization using the GCC-PHAT metric is applied to the two spectrograms $\mathbf{V_1^c}$ and $\mathbf{V_2^c}$ as explained in section 3.4.3. However, instead of processing several frames at once, it is better to obtain estimates for speaker locations at the frame level. In order to do so, one angular spectrum is needed per frame. Instead of summing the non-linear transformation of the GCC-PHAT metric $\mathbf{M}$ over all frequencies and subsequently maximizing the result over all frames as explained in section 3.4.3, o nly the summation over frequency bins is performed. As a result, the $N \times A$ localization matrix $\mathbf{L}$ with elements $l_{n\theta}$, defined in eq. (5.1), is obtained. Here N denotes the number of frames and A denotes at how many angles we evaluate the GCC-PHAT metric. It contains all angular spectra for all frames of the two mixture spectrograms.

$$l_{n\theta} = \sum_{f=1}^{F} m_{fn\theta} \tag{5.1}$$

The localization matrix is normalized so that all values lie in the range of 0 to 1. Peaks in $\mathbf{L}$ correspond to sound sources, which are in this setting speakers. In figure 5.2a, $\mathbf{L}$ is shown for a segment containing two speakers. However, one problem that arises when using the metric $\mathbf{L}$ is that $\mathbf{L}$ contains spurious peaks, i.e. false peaks which do not correspond to actual speakers. Such spurious peaks are more prominent for environments with a lot of reverberation. In order to limit the effect of these spurious peaks, an additional smoothing over time and space is necessary. Such smoothing is accomplished by applying a *Savitzky-Golay* filter in two dimensions; space and time [58]. The parameters of this filter are the window length w_{SG}

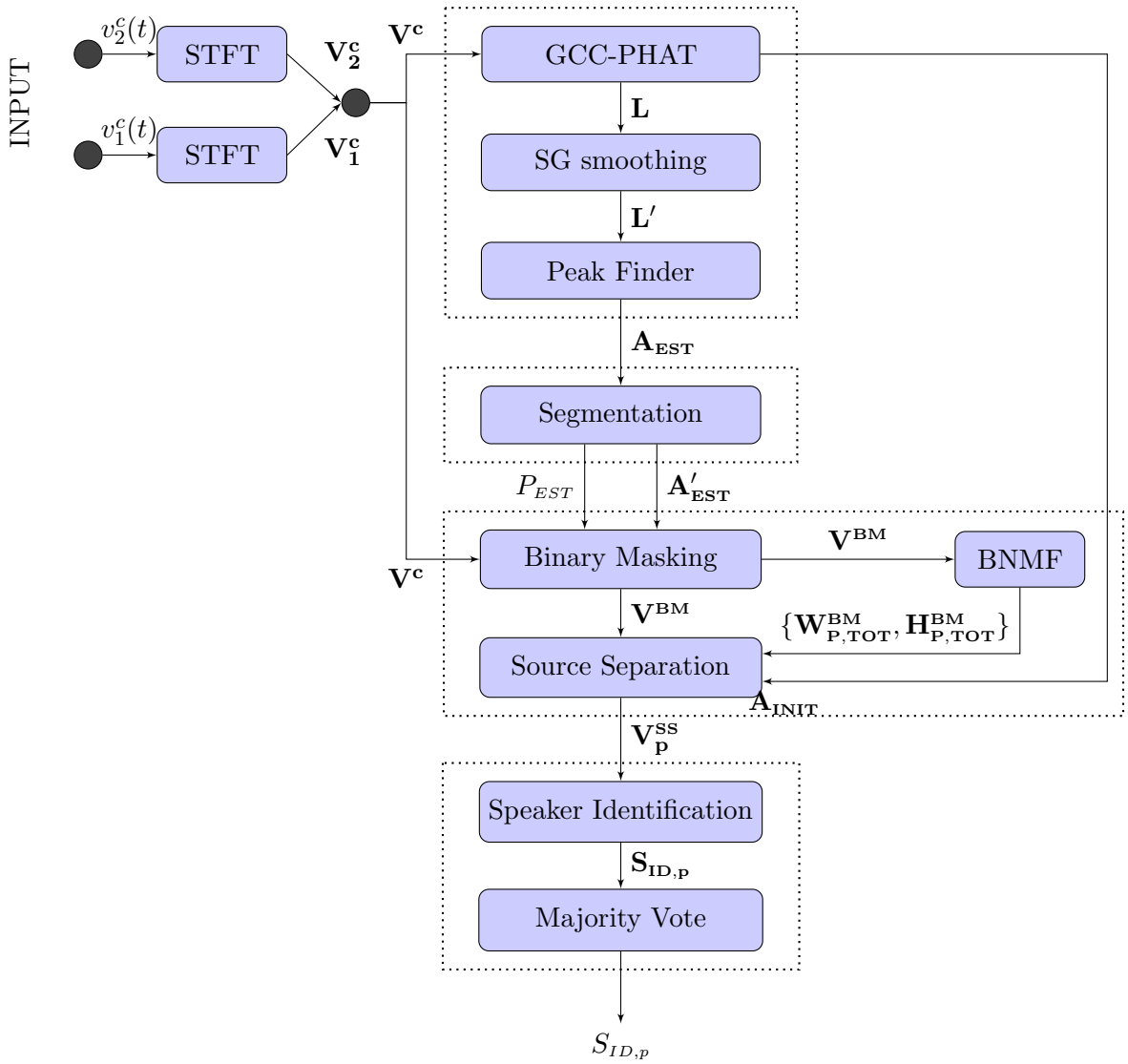


Figure 5.1: Schematic of joint speaker localization, enhancement and speaker identification

and the order of the filter k_{SG}. The window length indicates how many data points are taken into account when smoothing. For each data point, the Savitzky-Golay filter fits a low-order polynomial to the data points in a window around the data point [58]. This is why the window length should be odd. The value of this polynomial at the data point is then assigned as the new data point value. The order of the filter is the order of the polynomial that is fitted to the context window. It is a trade-off between distortion and noise reduction [59]. Higher order Savitzky-Golay filters tend to distort the signal less, but do not filter away high-frequency peaks. Lower order Savitzky-Golay filters will filter away more peaks, but they distort the signal more severely and any remaining peaks will be less prominent. As a consequence, peaks corresponding to true speakers might be filtered away if the order is too low. An advantage of the Savitzky-Golay filter is that it can be implemented as a (fast) FIR convolution filter. Here, we set the order to 1 and the window length to 15 for time smoothing. The order for time

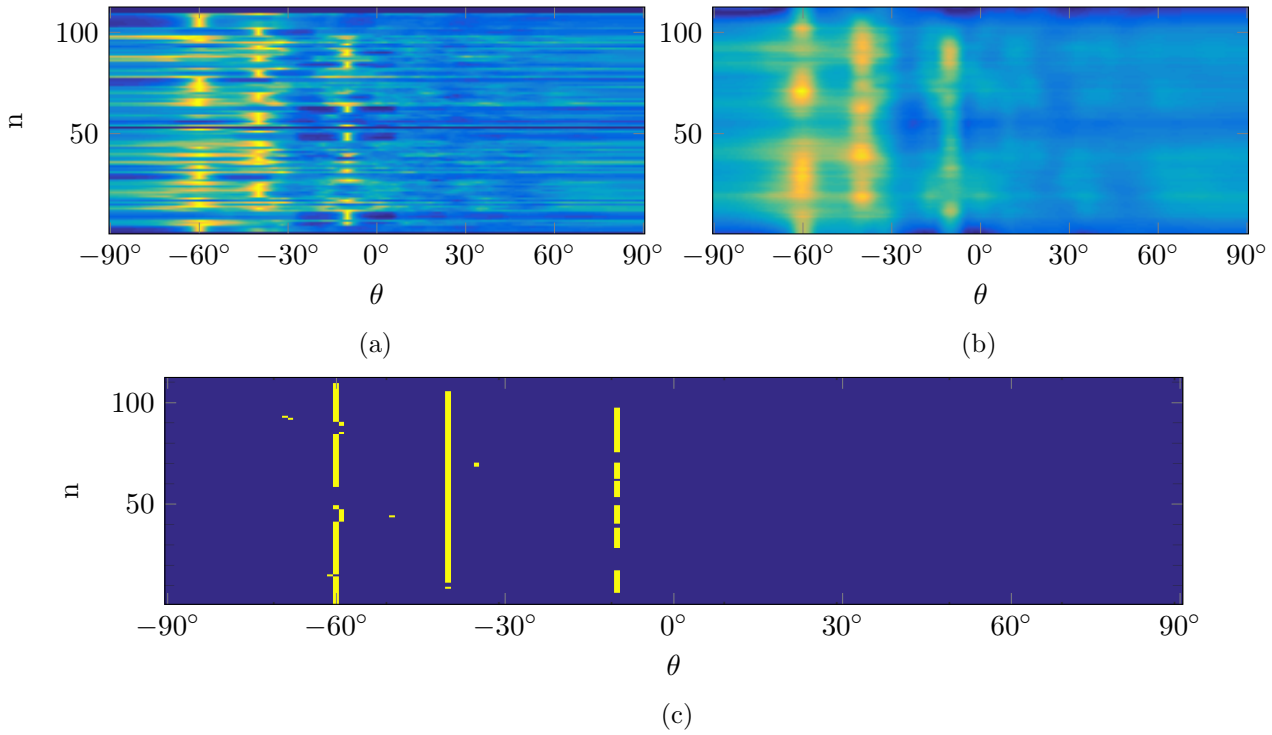


Figure 5.2: Localization of a speech segment of 3.57s with three simultaneous speakers; at $-60°$, $-40°$ and $-10°$. (a) $\mathbf{L}$ is the localization matrix contain the angular spectrum for each frame. (b) $\mathbf{L}'$ is the smoothed version of $\mathbf{L}$ and contains fewer spurious peaks caused by e.g. reverberation. (c) The boolean matrix $\mathbf{A}_{\mathbf{EST}}$ containing the spatiotemporal sound source detections from a peak finding algorithm applied to $\mathbf{L}'$.

smoothing can be set to a low value because we expect a speaker to speak longer. However, the order for spatial smoothing should be set to a higher order since spatial peaks can be sharp. We suggest to set the order to 4 and the window length to 10 for spatial smoothing. The smoothed version of the localization matrix is denoted by the symbol $\mathbf{L}'$. In figure 5.2b, an example of such a smoothed localization matrix can be found.

A peak finding algorithm is applied to each row of $\mathbf{L}'$. The most important parameters of the peak finding algorithm are the *minimum peak separation* and the *minimum peak height*. These parameters need to be set according to assumptions about the scenario where localization will be applied. For each frame n, the angles $\theta_{i,n}$ where a peak has been found are stored. This spatiotemporal information can be represented intuitively in a matrix $\mathbf{A}_{\mathbf{EST}}$. This boolean matrix has the same dimensionality as $\mathbf{L}'$ and contains a 1 if an angle has been detected and a 0 elsewhere. An example of such a matrix for the same case of figure 5.2b can be found in figure 5.2c.

5.2.3 Spatiotemporal segmentation

Speaker localization produces a set of per-frame sound source detections and corresponding bearing angles. These spatiotemporal detections corresponding to a scene are respresented in the $N \times A$ matrix $\mathbf{A}_{\mathbf{EST}}$; for a given frame n and angle θ, a 1 indicates that a source has been

detected and a 0 indicates that no source has been detected. During segmentation, the goal is to group such detections over time and space into single-speaker segments. This is a crucial step since speaker identification is impossible if multiple speakers are active in the same segment (see chapter 4). Unlike conventional segmentation, the spatial information is considered as well in this setting. For example, instead of indicating that

"Segment p starts at frame n_1 and stops at frame n_2.",

we indicate that

"Segment p starts at frame n_1 and stops at frame n_2 and the speaker speaking in this segment is located at angle A_1.".

As a result, it is possible for segments to overlap, i.e. the system can deal with speakers who speak simultaneously.

However, as can be seen in figure 5.2c, localization is inexact and this complicates the segmentation process. Several reasons may cause these incorrect sound source detections. First of all, not every spurious peak caused by reverberation is filtered out by smoothing. Evidently, since they are not removed, they are detected by the peak finding algorithm and result in false spatiotemporal sound source detections. Secondly, it is possible that the far-field assumption does not hold. That is, when sound sources are not sufficiently far from the microphone array. In that case, the difference between both channel recordings of the direct sound, i.e. the sound reaching the microphones without being reflected first, can no longer be modelled by a delay which is only dependent on the bearing angle relative to the axis of the microphone array. If the far-field assumption does not hold, the delay is dependent on the distance between the sound source and the microphone array as well and detections may be off by a few degrees. Finally, when using the GCC-PHAT metric, sound sources located at distinct positions should be independent of each other. Since the GCC-PHAT metric is based on correlation, different sound sources might interfere with each other if sound sources are correlated. The problem worsens in the case of reverberation. Reverberation can be modelled by a series of many short-time delays. As a consequence, the probability of two highly correlated sounds produced by two different speakers occurring simultaneously increases if the reverberation time increases.

No matter the cause of these erroneous estimates, it is possible to group speaker detections across time and space and subsequently distinguish the correct from the erroneous detections by using high-level assumptions. The segmentation process consists of two steps: intersegmental separation and intrasegmental rejection.

Intersegmental separation consists of clustering spatiotemporal speaker detections according to some high-level assumptions. Many assumptions can be used to distinguish disjoint segments. In this work, the following assumptions are made:

1. **Minimum angular separation $\Delta_{A,min}$**
   Speakers are assumed to be located at different angles relative to the axis of the microphone array and the difference between these angles should be sufficiently large. That is, the angular separation should be larger than $\Delta_{A,min}$. If this is the case, spatiotemporal speaker detections are assigned to different segments and vice versa.
2. **Maximum intersegmental silence $\Delta_{N,max}$**
   If multiple spatiotemporal speaker detections occur at the same bearing angle, it is possible that these all belong to the same speaker. However, if silence occurs between the first and the last detection, it is possible that an initial speaker has moved to another position during this silence and that another speaker has moved to the initial bearing angle of the initial speaker. Therefore, if the silence between two spatiotemporal speaker detections

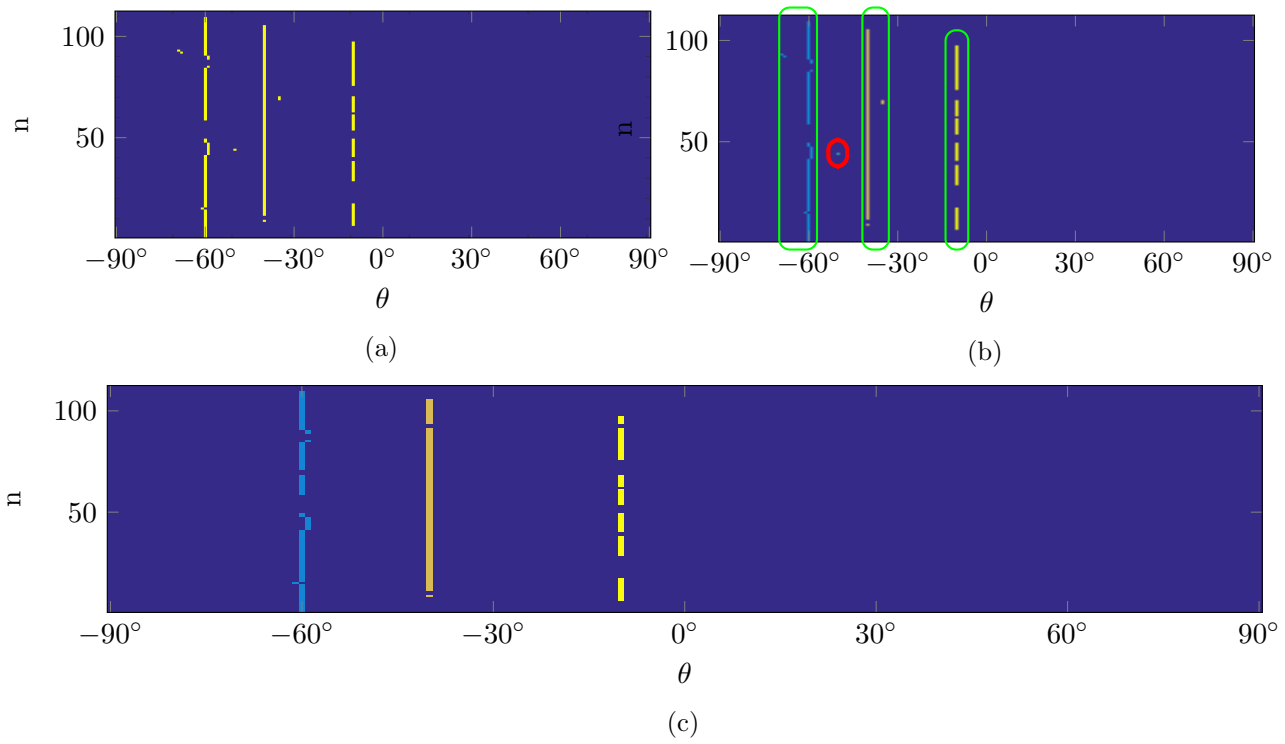


Figure 5.3: Segmentation for the segment with three simultaneous speakers from figure 5.2. (a) Spatiotemporal speaker detections in $\mathbf{A}_{\mathbf{EST}}$ are obtained from localization. (b) Intersegmental separation is applied to $\mathbf{A}_{\mathbf{EST}}$ and several clusters are obtained corresponding to single-speaker segments. (c) Intrasegmental rejection eliminates any remaining erroneous speaker detections.

expressed in frames is larger than $\Delta_{N,max}$, these spatiotemporal speaker detections are assigned to separate segments and vice versa.

Figure 5.3b shows how spatiotemporal speaker detections shown in figure 5.3a can be clustered into segments using intersegmental separation. The estimated segments shown in figure 5.3b have been boxed; green boxes indicate frames that correspond to actual speakers, red boxes correspond to erroneous segments. For this case, four segments were found. First, it can be seen that for some frames, multiple simultaneous speaker detections can be found within a segment, contradicting the assumption of single-speaker segments. For example, in the blue segment in figure 5.3b, there are multiple spatiotemporal detections at n=95. Secondly, it can be seen that some segments are extremely short, e.g. the erroneous segment with at angle 50°.

Again, high-level assumptions can be used to rectify these wrong detections. The following assumptions can be used to improve localization and segmentation results:

1. **Minimum utterance length $\Delta_{N,min}$**
   When a segment is too short, i.e. shorter than $\Delta_{N,min}$ when expressed in frames, it is assumed to be wrong and the segment is disregarded.
2. **Maximum one speaker per segment**
   Given the assumption that speakers are separated by a minimum angular separation $\Delta_{A,min}$, it is impossible that within a segment and within a frame multiple spatiotemporal speaker detections occur. If such simultaneous intrasegmental speaker detections do occur, speaker detections need to be rejected until at most one intrasegmental speaker

detection remains per frame. To determine which detection should be kept, the mode of the bearing angle at which speaker detections occur is evaluated for each segment. If within a frame and a segment multiple speaker detections occur, only the speaker detection which lies closest to the mode is kept. For example, the multiple simultaneous spatiotemporal speaker detections in the blue segment from figure 5.3b at frame $n = 95$ have been eliminated with this rule.

Figure 5.3c illustrates the effect of intrasegmental rejection of spatiotemporal speaker detections when the assumptions stated above are used. The final result is a clean segmentation. The number of estimated segments in a scene is denoted by P_{est}. The remaining spatiotemporal source detections, appended with corresponding segment indices, are represented in the matrix $\mathbf{A'_{EST}}$.

5.2.4 Binary masking

As explained in chapter 3, EM source separation needs initial estimates for the parameters: the dictionary $\mathbf{W}$, the activation matrix $\mathbf{H}$, the source spectrograms $\mathbf{V}$ and the mixing matrix $\mathbf{A}$. An initial estimate for $\mathbf{A}$ is produced during the localization stage. Here, initial estimates for the complex source spectrograms $\mathbf{V_p^{BM}}$ for each detected segment p are produced using binary masking as explained in 3.4.3. In order to do so, we assume that a scene is static, i.e. speakers do not move in a scene during a recording. As such, The maximum intersegmental silence can be set to ∞.

The resulting set of initial estimates of the complex source spectrograms is denoted by

$$\mathbf{V^{BM}} = \{\mathbf{V_1^{BM}}, .., \mathbf{V_{P_{est}}^{BM}}\} \tag{5.2}$$

where P_{est} is the estimated number of concurrent single-speaker segments during the scene as determined during segmentation.

5.2.5 BNMF and enhancement (EM source separation)

For each initial estimate of the source spectrogram $\mathbf{V_p^{BM}}$ corresponding to a segment p, a factorization is computed. The resulting dictionaries $\mathbf{W_p^{BM}}$ and activation matrix $\mathbf{H_p^{BM}}$ are used as initial estimates for the EM source separation algorithm. All segment dictionaries are concatenated into a combined segment dictionary $\mathbf{W_{P,TOT}^{BM}}$ and all segment activation matrices are concatenated into a combined activation matrix $\mathbf{H_{P,TOT}^{BM}}$. The sets of objects that belong to different segments need to be given as an additional input to the EM source separation algorithm as well. The goal of EM source separation is to obtain better estimates of the source spectrograms. The separated spectrogram that EM source separation produces for a segment p is denoted by $\mathbf{V_p^{SS}}$ and the set of all separated spectrograms from a scene is denoted by $\mathbf{V^{SS}}$.

5.2.6 Object-based speaker identification

Assuming that source separation has been performed successfully, each separated spectrogram $\mathbf{V_p^{SS}}$ contains but a single speaker. Identification is now performed on each of these separated spectrograms. As a result, a set of features are obtained for each segment. These features are classified. Majority voting is applied to the resulting set of identity estimates $\mathbf{S_{ID,EST}}$ to produce the final speaker identity estimate $S_{ID,EST}$ corresponding to a segment.

5.3 Experiments and results

In order to evaluate the performance of joint speaker localization, enhancement and identification, we have conducted two experiments; one experiment has been conducted on non-reverberated signals, and another has been conducted on reverberated signals. In both experiments, speaker models were learned on non-reverberated single-channel signals.

Experiment 1: Joint speaker localization, enhancement and identification on non-reverberated recordings

The first experiment aims to demonstrate that our system is able to identify separated recordings in a non-reverberated setting. Such identification is not straightforward since separated signals often still have artefacts which have been wrongly separated. These artefacts can be seen as noise that complicate the identification process. The effectiveness of the localization and source separation modules have already been demonstrated by Mirzaei *et al.* [2]. We here show that this system can successfully be combined with object-based speaker identification as proposed in this thesis if the scenes are non-reverberated.

The experiment has been performed on signals from the CHiME database. Each simulated *scene* consists of multiple speakers who speak simultaneously. Speakers are located at different angles θ_s. The task in this experiment is to count and locate each speaker, determine when each speaker speaks, obtain the source spectrograms and subsequently identify each speaker in that order. The performance is measured by several indicators. For localization, we shall indicate if the correct count of speakers has been found. For source separation, three commonly used performance measures for source separation have been evaluated: Signal-to-Distortion Ratio (SDR), Signal-to-Interference Ratio (SIR) and Signal-to-Artifact Ratio (SAR) [60]. These performance measures have been used by Mirzaei *et al.* as well [2]. For identification, performance is measured by the fraction of detected segments which have correctly been identified.

The experiment has been performed as follows. The size of the set of known speakers is still denoted by S. The number of speakers who speak simultaneously during a scene is denoted by P. For each scene, P speaker identities are chosen randomly from the set of known speakers. It is impossible for the same identity to occur at multiple angles during a scene. For each of these identities, an utterance is chosen randomly that has not been used for training the speaker model. The corresponding spectrogram $\mathbf{V_s}$ is computed for these utterances. For each utterance, a bearing angle is chosen randomly from a set of angles in the range of $-60°$ to $60°$ with intervals of $10°$. Additionally, angles used in a scene differ by at least $20°$. The target bearing angles are simulated with pan pot mixing, i.e. the channel mixing matrices $\mathbf{A}$ for the simulated angle θ_s (see eq. (3.47)) are multiplied with the complex source spectrogram $\mathbf{V_s}$. This is equivalent to adding a pure time delay between both channels. The distance between the virtual receivers of the microphone array is set to 0.15 meters. No additional noise was added in these simulated scenes.

Once a scene has been simulated, the angles are extracted per frame using the non-linear GCC-PHAT metric. Segmentation is subsequently performed to determine which sound source detections can be grouped into single-speaker segments. The minimum angular separation $\Delta_{A,min}$ has been set to $8°$, the maximum intersegmental silence $\Delta_{N,max}$ has been set to ∞ and the minimum utterance length $\Delta_{N,min}$ has been set to 21 frames or 0.7 seconds. Since the scene is assumed to be static, we can safely say that no two speakers will occur at the same angle

during a scene, which is why the maximum intersegmental silence can be set to infinity. The initial estimates $\mathbf{V_p^{BM}}$ are subsequently factorized using NMF with Bayesian inference. The set of initial estimates is used to separate the mixture spectrogram using EM source separation. The resulting separated source spectrograms are denoted by $\mathbf{V_p^{ss}}$. Finally, object-based speaker identification is performed on each of the separated spectrograms $\mathbf{V_p^{ss}}$. The resulting set of features per segment is classified. To obtain the final speaker identity estimate $S_{ID,EST}$ for a segment, majority voting is performed on the set of features corresponding to that segment.

This experiment was performed four times; for two values of the known speaker set size S (4 and 8) and for two values of the number of simultaneous speakers during a scene P (2 and 3). Each set of parameters has been evaluated on 250 simulated scenes. The performance results for localization and identification are summarized in table 5.1. As previously mentioned, there are 250 scenes with P speakers per scene. The success rate for localization is expressed as the proportion of found speakers and the total number of speakers. The success rate for identification is expressed as the proportion of correctly identified speakers and the total number of found speakers. The performance results for source separation can be found in figure D.1 and table 5.3. Figure D.1 in appendix D shows the sorted source separation performance measures SDR, SIR and SAR for all separated source spectrograms $\mathbf{V_p^{ss}}$. The average SDR, SIR and SAR are given in table 5.3.

Experiment 2: Joint speaker localization, enhancement and identification on reverberated recordings

A second experiment has been conducted to illustrate the effect of reverberation on joint speaker localization, enhancement and identification. This experiment is identical to experiment 1, except for the spatial simulation of the signals. Instead of adding a pure delay, i.e. emulating a room without reverberation, each utterance is convoluted with a room impulse response which is specific to a certain source and receiver position. The room impulse responses have been generated using the roomsim packages. Figure B.1 in appendix B shows a list of possible source positions. The reverberation time RT_{60} differs for each source receiver pair, but it is approximately 0.280 seconds. This experiment has been performed for the same four parameter settings as experiment 1. The localization and identification results can be found in table 5.1 and the source separation results can be found in table 5.3 and in figure D.1 in appendix D.

Results

The results for localization and identification can be found in table 5.1. We would like to add that for $S = 4$ and $P = 2$ in the non-reverberated case, 6 of the 7 wrong identifications were caused by an unstability in the algorithm provided by Ozerov *et al.* which resulted in `NaN` values. We believe that this error happens because in some cases during source separation, the inverse of the correlation matrix for a given frequency bin $\mathbf{R_{ss,f}}$ is needed for updating the mixing matrix $\mathbf{A_f}$. In some cases, this matrix is singular which causes the error. However, we do not know why this correlation matrix is singular. This error also occured for $S = 4$ and $P = 3$ in the non-reverberated case (6 segments) and for $S = 4$ and $P = 3$ in the reverberated case (2 segments).

In the non-reverberated case and for $P = 2$, localization and identification is near perfect. However, identification performance degrades as P increases. We believe that this is because source separation is more difficult for higher P. As a consequence, the resulting separated

spectrograms $\mathbf{V}_{\mathbf{p}}^{\mathbf{ss}}$ are more noisy and identification of these noisy spectrograms leads to more errors. For the reverberated case, identification has the most errors. We observe that source separation is even worse in this case and thus leads to worse identification estimates (see next paragraph). It is also possible that the performance of our identification algorithm degrades if applied to reverberated signals. Unfortunately, we do not have any results to verify this.

The results for source separation can be found in table 5.3 and more detailed results can be found in appendix D. Source separation performance results have only been published for scenes where P_{EST} is correct. Two trends can be noticed. First, source separation performance is better for experiment 1 than for experiment 2. This can be due to the inaccurate count of speakers within a segment which occur more in experiment 2. As a result, the source separation algorithm is initialized incorrectly and will lead to wrong source separations. Secondly, source separation is more difficult if the number of simultaneous speakers P is higher.

5.4 Conclusion

In this chapter, we have shown how our method for object-based speaker identification system can be combined with existing techniques for localization and blind source separation. With a future real-time application in mind, our system incorporates localization at the frame level. Additional segmentation groups these spatiotemporal speaker detections into single-speaker segments according to high-level assumptions. As a result, identification can be performed on single-speaker separated segments.

We have shown that our system performs as expected in non-reverberated settings with two concurrent segments. However, performance degrades for scenes with three concurrent segments, as a result of an interaction between errors from different modules. Since less speakers are detected during localization and segmentation, source separation is initialized wrong and identification is performed on poorly separated spectrograms. As can be seen in appendix D, the performance results for blind source separation for a set of parameters contain a lot of variation. However, we believe that additional care in selection of the parameters of segmentation may lead to better localization results and better overall performance.

We have also shown that the performance of our system is affected by the amount of reverberation present in a scene. Localization is more difficult because reverberation leads to additional peaks in the angular spectrum which hinders speaker detection. As a result, the performance of blind source separation and speaker identification is lower as well. In future work, the interaction between these modules and the relative effect of reverberation on each of these modules should be studied more in-depth.

Exp.	S	P	Tot. # Scenes	Tot. # Speakers	Found speakers	Correct ID
1	4	2	250	500	490 (98%)	483(98.6%)
	4	3	250	750	687 (91.6%)	624 (90.8%)
	8	2	250	500	497 (99.4%)	496 (99.8%)
	8	3	250	750	714 (95.2%)	605 (84.7%)
2	4	2	250	500	487 (97.4%)	408 (83.8%)
	4	3	250	750	693 (92.4%)	529 (76.3%)
	8	2	250	500	491 (98.2%)	379 (77.2%)
	8	3	250	750	701 (93.5%)	458 (65.3%)

Table 5.1: Performance results for localization and identification for the case of non-reverberated recordings. Each row contains the number of correctly localized and correctly identified segments for the (1) non-reverberated and (2) reverberated experiment with parameters S and P.

			P_{EST}						
Exp.	S	P	1	2	3	4	5	6	Total
1	4	2	10	240					250
	4	3		63	187				250
	8	2	3	247					250
	8	3	1	34	215				250
2	4	2	13	230					250
	4	3	10	37	167	33	5	1	250
	8	2	9	230	9	2			250
	8	3	8	33	187	22			250

Table 5.2: Detailed results for localization. Each row contains the number of scenes with a specific P_{EST} for the (1) non-reverberated and (2) reverberated experiment with parameters S and P.

Exp.	S	P	Av. SDR	Av. SIR	Av. SAR
1	4	2	17.08	22.26	19.21
	4	3	4.04	7.68	7.87
	8	2	17.50	22.60	19.55
	8	3	4.24	7.83	8.12
2	4	2	9.04	13.68	11.41
	4	3	3.25	6.76	7.27
	8	2	9.47	14.18	11.72
	8	3	2.90	6.26	7.18

Table 5.3: Average source separation performance measures for the case of non-reverberated recordings.

Chapter 6

conclusion

In this thesis, our goal has been to develop a system for joint speaker localization, enhancement and identification for use in the Cametron project. To achieve this, techniques for factorization-based source separation and cross-correlation-based localization have been adopted. Additionally, we have proposed a new method for object-based speaker identification.

Speaker models are built on dictionaries $\mathbf{W_s}$ which are extracted via nonnegative matrix factorization from a spectrogram $\mathbf{V_s}$. These dictionaries contain sounds which are characteristic to speakers. Key issues such as model order selection, i.e. the choice of the number of objects by which we want to represent a speaker, and feature extraction have been addressed. Our proposed method has clear advantages compared to traditional methods. For instance, our system does not require a universal background model. As a result, our system is applicable when very little data is available and is not affected by the composition of the used training data. Additionally, we believe that extraction of characteristic sounds is a more natural way of modeling speakers. However, our system is meant for applications where the set of known speakers is not large, i.e. smaller than 15, because feature dimensionality increases linearly with the number of known speakers. We have shown that our system performs excellently when each speaker model is trained on 5 minutes of high quality recordings. Additionally, we have shown that our system performs adequately on noisy field recordings; a first step toward use within real life environments. However, performance can be boosted if labeling of the training data is done with more care.

Finally, we have evaluated speaker identification in a system with increased functionality. By applying speaker identification on separated signals, speaker identification can be combined with localization and enhancement. Localization proves to be troublesome when applied to reverberated systems. As a result, the performance of the joint system performance deteriorates.

There is still room for improvement. Therefore, we would like to mention some possible directions for future research. First, application of a mel filter bank on $\mathbf{V_s}$ should be examined. Such a transformation may drastically reduce the bins by which sounds are represented. As a result, computational complexity might decrease drastically. However, the effect on source separation should be examined; the mel transformation can be reversed, but the resulting spectrogram is an approximation of the original spectrogram. Secondly, model order estimation is performed independently from Bayesian factorization. In future work, the derivation of extensions to nonnegative matrix factorization with Bayesian inference that incorporate relevance parameters for model order estimation should be investigated. Thirdly, our system requires that speakers are trained in advance. Extensions incorporating online learning can be implemented.

Online extensions of nonnegative matrix factorization and support vector machines exist. If such an implementation can be achieved, speakers could be learned on the fly. Additional information supplied visual sources could aid such online algorithms in a semi-supervised way. Finally, techniques should be examined that account for reverberation. Such techniques include dereverberation by deconvolution or training on reverberated training sets.

Appendix A

Data sets

In chapter 4, it is shown how NMF can be used within a speaker identification system, and in chapter 5, this speaker recognition system is extended with speaker bearing detection and speaker source separation. To evaluate the performance of speaker recognition and source separation, a data set containing a large number of audio files spoken by a sufficiently large amount of speakers is needed for training speaker models and validation of the methods. Obviously, these audio files need to be labelled with the correct speaker identity. In this thesis, two data sets have been used; the CHiME corpus and a set of recorded panel meetings.

A.1 Data set 1: CHiME corpus

Grid Corpus

In order to evaluate a speaker identification method, a data set of labelled utterances from numerous speakers is needed. For the task of language independent speaker identification, the content of each utterance is irrelevant. Only the identity of the correct speaker identity is necessary.

One suitable data set is the **CHiME** database, which is based on the **Grid corpus** [61, 62]. The Grid corpus consists of utterances from a total of 34 speakers; 18 male speakers and 16 female speakers. Originally, the motivation for the development of the Grid corpus has been to provide a sufficiently large data set for audio-visual automatic speech recognition. In this work, only the auditive part of the corpus is useful.

All utterances were originally recorded with a sample frequency of 50 kHz and subsequently downsampled to 25 kHz [62]. Speakers were asked to stand in an acoustic booth and read aloud a series of 1000 syntactically simple questions. These sentences have the following fixed structure of components; a *command* followed by *a color*, *a proposition*, *a letter*, *a digit* and *an adverb*. A listing of the possible parameter values used for each component in the Grid corpus is provided in table A.1 [62]. The combination of the words for each sentence is chosen randomly and thus there are no linguistic cues that can be used for identification. The speakers were given $1.5 - 3$ seconds to record each sentence. The entire Grid corpus contains 34000 utterances.

Component	# Possible values	Parameter values
Command	4	bin, lay, place, set
Color	4	blue, green, red, white
Preposition	4	at, by, in, with
Letter	25	A-Z (without W)
Digit	10	0-9
Adverb	4	again, now, please, soon

Table A.1: Structure of the spoken Grid utterances

Chime corpus

As mentioned before, the Grid corpus has been recorded in an acoustic booth. Although limited, there is some reverberation effect present in these recordings. For the CHiME corpus, a compensation filter has been applied to cancel the effect of the acoustic booth [61]. This was initially done to obtain recordings that could be convoluted with a room impulse response to emulate the room characteristics of an environment where natural noise was recorded. This way, natural sounding recordings with controlled sound-to-noise ratios can be obtained [61]. However, we are not interested in noisy environments. Both clean and noisy utterances are provided in the CHiME corpus. However, only the clean data set is used in this thesis. These utterances are suited for the spatial simulation (see appendix B). The clean utterances provided in the CHiME database have been downsampled to 16 kHz.

Motivation

The Grid corpus has been recorded for use in automatic speech recognition and speech source separation. However, we believe that the use of this corpus is fit for evaluating our speaker recognition system as well. Firstly, this data set contains utterances from 34 speakers, which is sufficient for both training and evaluation of the speaker recognition system. Secondly, each speaker speaks the same words from the limited dictionary given in table A.1. This eliminates the possibility that speakers are recognized because of speaker-dependent vocabulary. Thirdly, the reverberation of the acoustic booth has been cancelled out, resulting in audio signals that are are not affected by a room's reverberation. This is necessary because these utterances will need to be convolved with a room impulse response as explained in appendix B.

One disadvantage of the Grid corpus is its limited vocabulary. In a real-life setting, it is implausible that each speaker only uses a limited set of words, unless speakers only give commands from a predefined set. Due to the limited vocabulary, speaker models will be trained which focus on the most occurring sounds in this vocabulary. The developers of the Grid corpus have taken special care to account for this by phonetically balancing the corpus. On the one hand, they have added the *'Letter'*-component and, on the other hand, for the other components they have chosen parameter values that contain all different phonetic classes.

Despite this, we assume that performance of speaker models trained using the Grid corpus will not work well when applied to natural recordings of conversations. To account for this, a secondary database has been used to test the speaker recognition system with natural recordings of academic panel meetings. More information about this database is given in appendix A.2.

Figure A.1: The setup of the panel meeting. Ten speakers are attending the panel meeting; eight male speakers and two female speakers.

ID tag	M1	M2	M3	M4	M5	M6	M7	M8	F1	F2
Gender (M/F)	M	M	M	M	M	M	M	M	F	F
Duration (s)	1991	1509	660	1653	277	590	1103	918	681	56

Table A.2: Speaker information for the panel meeting data set

A.2 Data set 2: Panel meeting database

The CHiME database has been recorded in a studio in well-controlled conditions. These recordings are of high quality. In a typical real-life situation, it is improbable that such ideal recordings are available. It is therefore useful to have an additional data set which has been recorded in such a typical real-life setting. One such situation where speaker identification is applicable is a panel meeting.

A data set of a typical panel meeting has been recorded by Punarjay Chakravarty. The data set comprises of a stereo audiovisual signal which has been recorded with a camcorder. The visual data is unnecessary for experiments performed in this thesis and has been disregarded. However, the audio signal provides an excellent opportunity to test speaker identification in real-life conditions. Unfortunately, the two camcorder microphones are too closely spaced for use in speaker localization.

The entire data set consists of 3 hours and 15 minutes of audiovisual data. The audio signal has been sampled at a sampling frequency of 144000 Hz. However, for use in speaker identification, these signals have been resampled at 16000 Hz. For use in this thesis, the entire data set has been labelled with the correct speaker identities. Special care has been taken to make sure that loud noises such as closing doors or coughing was labelled as non-speech. However, various more quiet noises such as squeaky chairs or occasional murmuring have been allowed as we believe that labelling in a real-life application will be imperfect as well.

There are a total of ten speakers. Naturally, not all speakers have talked for the same amount of time. A summary of speaker ID tags, gender and total duration of speech can be found in table A.2. Notice that the mentioned duration includes the natural pauses in spoken speech.

Appendix B

Spatial simulation

The evaluation of speaker bearing detection and source separation require recordings of conversations where multiple speakers which are spatially separated do or do not speak simultaneously. Due to limited resources, we have chosen to simulate these situations as realistically as possible. This approach also provides greater flexibility, which allows us to test a larger number of different situations.

B.1 Room impulse response

When listening to sound recordings, most people will be able to roughly guess the size and type of the room where the recording was made. The distinction can be made as a result of reverberation. The sound which reaches the microphone comes partly directly from the sound source and partly from reflections from the walls. Recordings made in dissimilar rooms sound different because of these reflections.

One way to characterize the acoustics of a room is by measuring its **room impulse response**. In essence, the RIR is nothing more than a measurement of the characteristic reflections of a room if an impulse at a sound source's position is heard by an observer. Since reflections are characteristic for both the location of the sound source and the location of the recording element, a RIR is dependent on both locations as well.

An approximation of the RIR can be measured by playing an approximation of a Dirac impulse, e.g. a gun shot or single clap, and recording the resulting sound. Such approximations are however imperfect. Accurate RIR measurements are possible using the SineSweep technique developed by Farina [63]. Other techniques for RIR measurement can be found in a paper by Stan *et al.* [64].

RIR's are important for testing speaker bearing detection, i.e. estimation of the azimuth of a speaker relative to a microphone array, and for testing the robustness of speech processing techniques in general. Unreverberated recordings such as those from appendix A can be convolved with a RIR to emulate the acoustics of a room. However, the key problem is that the measurement of RIR's can be tedious. Luckily, methods exist for modeling RIR's. High-end programs for simulating acoustic acoustics of architecturally complex structures are available. However, for the purposes in this thesis, i.e. analysing speaker recognition in the presence of moderate reverberation, modeling reverberation of simple rectangular rooms is sufficient.

B.2 Roomsim package

The Roomsim package, which has been developed by Campbell *et al.* in Matlab, enables simulation of 'shoebox' acoustics [65]. The code can be found online [66]. Roomsim is based on the image method developed by Allen *et al.* [67]. In short, reflections are modeled as virtual sources which are located beyond the boundaries of the room. As such, the room is duplicated endlessly and the location of the virtual sources is determined by mirroring the original sound source with respect to the walls. However, not all virtual sources, i.e. reflections, need to be taken into account. One can choose to limit the impulse response temporally if only virtual sources are considered within a certain radius of the original sound source. The maximum length L_{RIR} of the impulse response in samples is then

$$L_{RIR} = \frac{r}{v} \tag{B.1}$$

where r is the radius and v is the velocity of sound. It is also possible to limit the order of reflections, i.e. the maximum number of times sound can bounce of walls before arriving at the receiving sensor.

The Roomsim package is a great compromise between computational complexity and realistic sound modeling. Besides room size and RIR length, the user is provided with several other configurable parameters such as choice of surface material with frequency-dependent absorption coefficients, and sensor type and directionality [65]. Head related transfer functions are supported as well, but will not be used in this thesis as we assume that there is a microphone array present without a manikin emulating the body.

For the purpose of this thesis, a room with moderate reverberation has been modeled. We have only simulated a room with moderate reverberation because it is not our goal to provide a system that is designed for robustness. However, it is useful to investigate how the system is affected by moderate reverberation which occurs in most real-life scenes. The emulated room is of the following dimension; the width is 15 meters, the length is 12 meters and the height of the room is 4 meters. All absorption coefficients are equal to 0.75 for the entire frequency range and for all bounding surfaces. High absorption coefficients indicate weak reverberation and vice versa.

The origin of a coordinate system is placed in one of the corners of the rooms, as shown in figure B.1. For each of these rooms a set of two omnidirectional microphones are placed at coordinates (6 meters, 3.58 meters, 1.5 meters) and (6 meters, 3.43 meters, 1.5 meters). The microphones are separated by a distance of 0.15 meters. Several possible source positions are placed at a distance of 4 meters from the receiving sensor pair. Each possible source location is located at a different angle relative to the sensor pair. The convention adopted here is that the broadside direction corresponds with angle $0\,^\circ$ and the end-fire directions correspond with angles $-90\,^\circ$ and $90\,^\circ$. Possible source locations are located at angles between $-60\,^\circ$ to $60\,^\circ$ with intervals of $10\,^\circ$. In total there are 13 possible source locations, as can be seen in figure B.1. The Roomsim package produces impulse responses for each source-receiver pair [65]. Since there are 13 possible source locations and 2 receiving microphones in the microphone array, 26 RIR's are generated. Clean signals from the database (see appendix A) can be convolved with the two impulse responses corresponding to both receivers and a certain source position to obtain a stereo audio signal emulating a speaker speaking at that location and being recorded by the microphone array. The reverberation time differs slightly for each source-receiver pair. The minimum RT_{60} is equal to **0.263 seconds** and the maximum RT_{60} is equal to **0.301 seconds**.

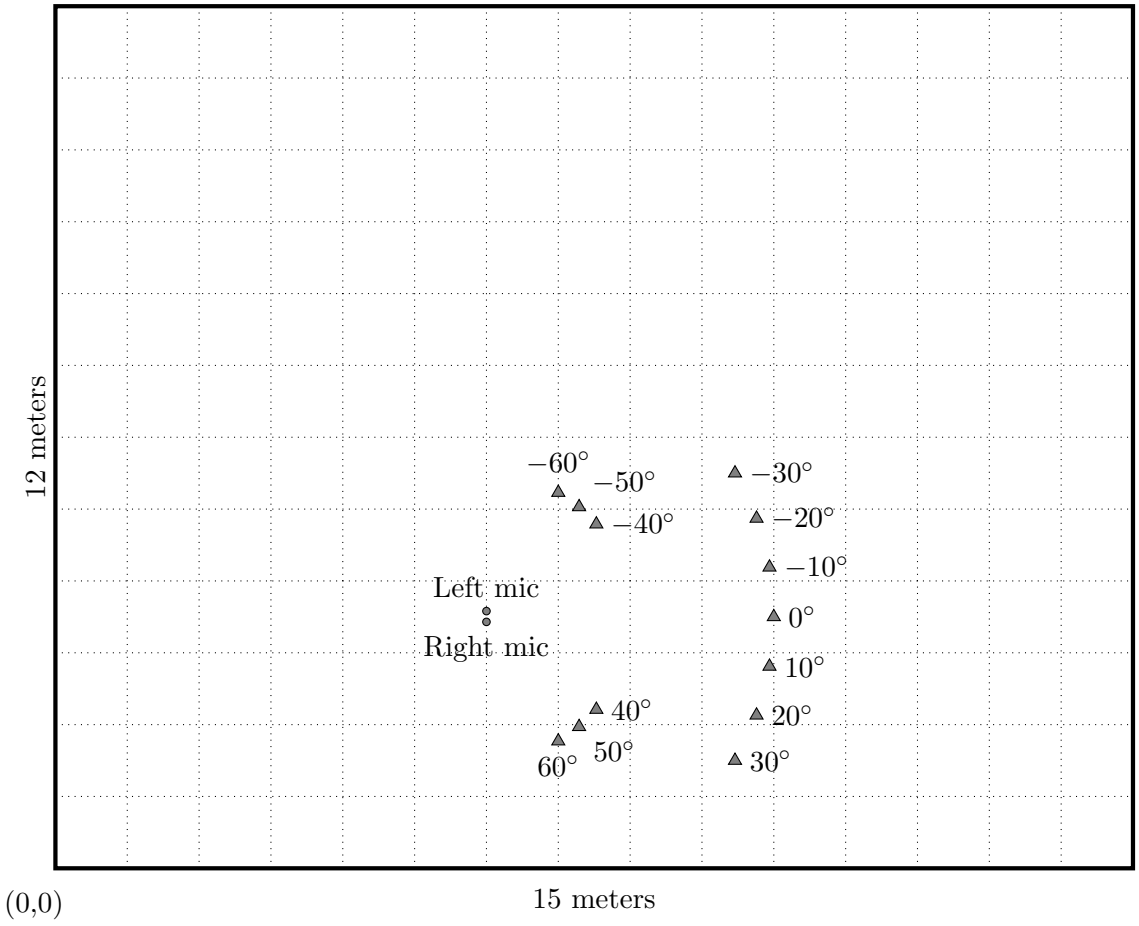


Figure B.1: Top view of simulated room with microphone array, possible sensor locations and possible source locations. Each possible source position is labelled with the angle with respect to the end-fire direction.

Appendix C

Algorithms

C.1 Algorithms for chapter 3

Algorithm 1: Nonnegative matrix factorization with Euclidean distance

Data: $\mathbf{V}^+, N_{iter}$
Result: $\mathbf{W}^+, \mathbf{H}^+$
for $n = 1 : 1 : N_{iter}$ **do**

$$\mathbf{H} \leftarrow \mathbf{H} \cdot \frac{\mathbf{W}^T\mathbf{V}}{\mathbf{W}^T\mathbf{W}\mathbf{H}};$$

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \frac{\mathbf{V}\mathbf{H}^T}{\mathbf{W}\mathbf{H}\mathbf{H}^T};$$

end

Algorithm 2: Nonnegative matrix factorization with Kullback-Leibler divergence

Data: $\mathbf{V}^+, N_{iter}$
Result: $\mathbf{W}^+, \mathbf{H}^+$
for $n = 1 : 1 : N_{iter}$ **do**

$$\mathbf{H} \leftarrow \mathbf{H} \cdot \frac{\mathbf{W}^T\dfrac{\mathbf{V}}{\mathbf{W}\mathbf{H}}}{\mathbf{W}^T\mathbf{1}^{F\times N}};$$

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \frac{\dfrac{\mathbf{V}}{\mathbf{W}\mathbf{H}}\mathbf{H}^T}{\mathbf{1}^{F\times N}\mathbf{H}^T};$$

end

Algorithm 3: Nonnegative matrix factorization with Itakura-Saito divergence

Data: $\mathbf{V}^+, N_{iter}$
Result: $\mathbf{W}^+, \mathbf{H}^+$
for $n = 1:1:N_{iter}$ **do**

$$\mathbf{H} \leftarrow \mathbf{H} \cdot \frac{\mathbf{W}^T \dfrac{\mathbf{V}}{(\mathbf{WH})^{.2}}}{\dfrac{\mathbf{W}^T}{\mathbf{WH}}};$$

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \frac{\left(\dfrac{\mathbf{V}}{(\mathbf{WH})^{.2}} \cdot \mathbf{V}\right)\mathbf{H}^T}{\dfrac{\mathbf{H}^T}{\mathbf{WH}}};$$

end

In algorithms 1 to 3, the operators $\mathbf{A} \cdot \mathbf{B}$ and $\frac{\mathbf{A}}{\mathbf{B}}$ respectively denote the elementwise product and division of two matrices $\mathbf{A}$ and $\mathbf{B}$. The operator $\mathbf{A}^{.2}$ denotes the element-wise squaring of a matrix $\mathbf{A}$.

The parameter N_{iter} determines how many times the set of update rules is applied. An alternative method is to measure the average or largest deviation ϵ of the elements of $\mathbf{W}$ and $\mathbf{H}$ between two successive iterations and quit the loop when ϵ drops below a threshold τ.

C.2 Algorithms for chapter 4

Algorithm 4: Automatic relevance determination

Data: $\mathbf{V}, a, b, K_{init}, N_{iter}$
Result: $\mathbf{W}, \mathbf{H}, \boldsymbol{\lambda}$
$[F, N] \leftarrow \text{size}(\mathbf{V})$;
$c \leftarrow F + N + a + 1$;
for $i = 1:1:N_{iter}$ **do**

$$\mathbf{H} \leftarrow \mathbf{H} \cdot \left(\frac{\mathbf{W}^T \dfrac{\mathbf{V}}{\mathbf{WH}}}{\text{repmat}\left(\text{sum}(\mathbf{W}, 1)^T + \dfrac{\varphi}{\boldsymbol{\lambda}}, 1, N\right)} \right);$$

$$\mathbf{W} \leftarrow \mathbf{W} \cdot \left(\frac{\dfrac{\mathbf{V}}{\mathbf{WH}} \cdot \mathbf{H}^T}{\text{repmat}\left(\text{sum}(\mathbf{H}, 2)^T + \dfrac{\varphi}{\boldsymbol{\lambda}^T}, F, 1\right)} \right);$$

$$\boldsymbol{\lambda} \leftarrow \frac{\text{sum}(\mathbf{W}, 1)^T + \text{sum}(\mathbf{H}, 2) + b}{c};$$

end

Algorithm 5: Block-based group sparsity nonnegative matrix factorization

Data: $\mathbf{V}, \mathbf{W_{TOT}}, N_{iter}$
Result: $\mathbf{H_{GS}}$
$\mathbf{\Lambda}_1 \leftarrow \lambda_1 \cdot \mathbf{1}^{K_{TOT} \times N}$;
$\mathbf{W}_{norm} \leftarrow \text{repmat}\left(sum(\mathbf{W}, 1)^T\right)$;
$\mathbf{H_{GS}} \leftarrow \mathbf{1}^{\mathbf{K_{TOT} \times N}}$;
for $i = 1:1:N_{iter}$ **do**
$\mathbf{H}_\Sigma \leftarrow sum(\mathbf{H_{GS}}, 2)$;
$\mathbf{\Lambda}_G \leftarrow \text{repmat}\left(\lambda_g \cdot \mathbf{H}_\Sigma \cdot \left(\mathbf{G}_B^T \sqrt{\mathbf{G}_B \mathbf{H}_\Sigma^{.2}}\right), 1, N\right)$;
$$\mathbf{H_{GS}} \leftarrow \mathbf{H_{GS}} \cdot \left(\frac{\mathbf{W_{TOT}}^T \dfrac{\mathbf{V}}{\mathbf{W_{TOT}} \mathbf{H_{GS}}}}{\mathbf{W}_{norm} + \mathbf{\Lambda}_G + \mathbf{\Lambda}_1} \right);$$
end

Algorithm 6: Frame-based group sparsity nonnegative matrix factorization

Data: $\mathbf{V}, \mathbf{W_{TOT}}, N_{iter}$
Result: $\mathbf{H_{GS}}$
$\mathbf{\Lambda_1} \leftarrow \lambda_1 \cdot \mathbf{1}^{K_{TOT} \times N}$;
$\mathbf{W}_{norm} \leftarrow \text{repmat}\left(sum(\mathbf{W_{TOT}}, 1)^T\right)$;
$\mathbf{H_{GS}} \leftarrow \mathbf{1}^{\mathbf{K_{TOT} \times N}}$;
for $i = 1:1:N_{iter}$ **do**
$\mathbf{\Lambda}_G \leftarrow \lambda_g \cdot \mathbf{H_{GS}} \left(\mathbf{G}_B^T \sqrt{\mathbf{G}_B \mathbf{H_{GS}}^{.2}}\right)$;
$$\mathbf{H_{GS}} \leftarrow \mathbf{H_{GS}} \cdot \left(\frac{\mathbf{W_{TOT}}^T \dfrac{\mathbf{V}}{\mathbf{W_{TOT}} \mathbf{H_{GS}}}}{\mathbf{W}_{norm} + \mathbf{\Lambda}_G + \mathbf{\Lambda}_1} \right);$$
end

Appendix D

BSS performance results

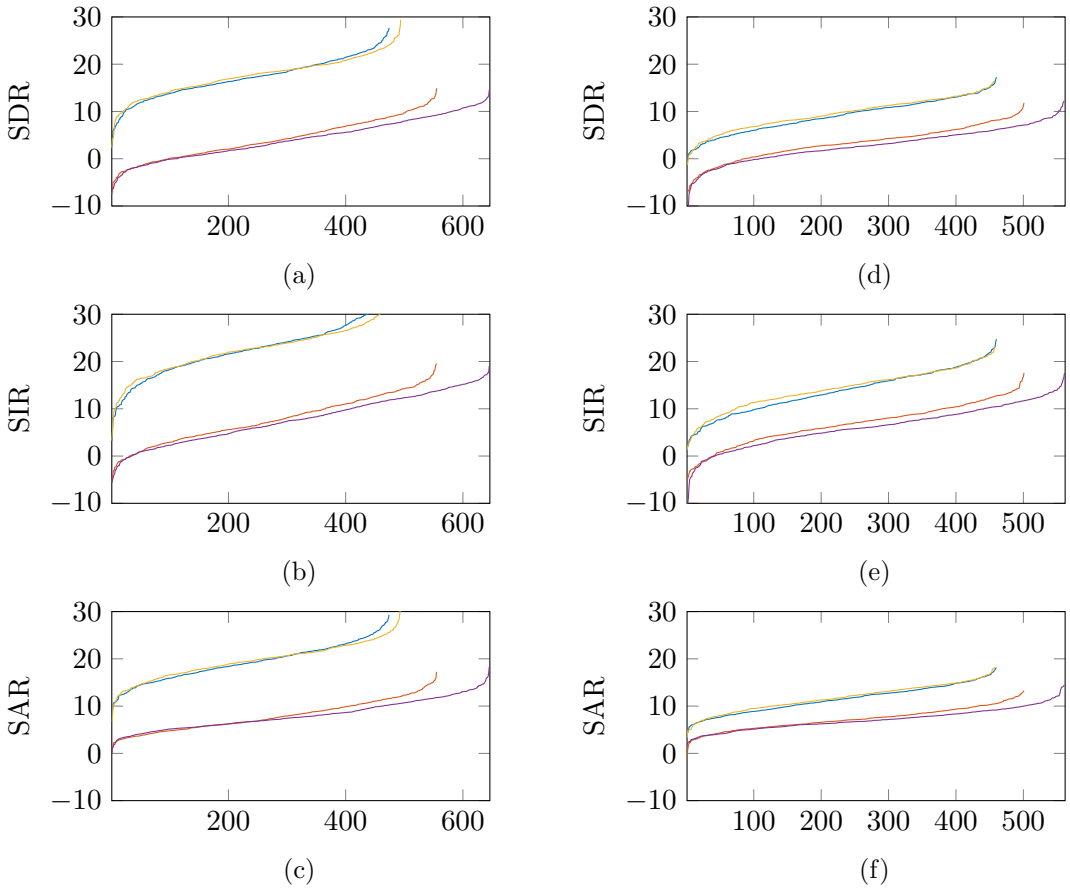


Figure D.1: Sorted source separation performance measures for four parameter settings. (a) Signal-to-Distortion Ratio, (b) Signal-to-Interference Ratio and (c) Signal-to-Artifact Ratio for non-reverberated signals and (d) Signal-to-Distortion Ratio, (e) Signal-to-Interference Ratio and (f) Signal-to-Artifact Ratio for reverberated signals. Color codes: Blue: $S = 4$ and $P = 2$, orange: $S = 4$ and $P = 3$, yellow: $S = 8$ and $P = 2$ and purple: $S = 8$ and $P = 3$.

Bibliography

[1] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 18, no. 3, pp. 550–563, 2010.

[2] S. Mirzaei, H. V. Hamme, and Y. Norouzi, "Blind audio source separation of stereo mixtures using bayesian non-negative matrix factorization," in *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European.* Lisbon,Portugal: IEEE, 2014, pp. 621–625.

[3] N. Singh, R. Khan, and R. Shree, "Applications of speaker recognition," *Procedia Engineering*, vol. 38, pp. 3122–3126, 2012.

[4] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.* IEEE, 2003, pp. 177–180.

[5] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based speech enhancement and its application to noise-robust automatic speech recognition," in *Proc. 1st Int. Workshop on Machine Listening in Multisource Environments (CHiME)*, Florence, Italy, 2011, pp. 53–57.

[6] A. Hurmalainen, R. Saeidi, and T. Virtanen, "Group sparsity for speaker identity discrimination in factorisation-based speech recognition." in *INTERSPEECH*, 2012.

[7] C. Joder and B. Schuller, "Exploring nonnegative matrix factorization for audio classification: Application to speaker recognition," in *Speech Communication; 10. ITG Symposium; Proceedings of.* VDE, 2012, pp. 1–4.

[8] "Cametron — esat ku leuven," http://www.esat.kuleuven.be/psi/visics/research/projects/CAMETRON, accessed: 2015-06-02.

[9] V. Y. Tan and C. Févotte, "Automatic relevance determination in nonnegative matrix factorization with the/spl beta/-divergence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 7, pp. 1592–1605, 2013.

[10] H. Beigi, *Fundamentals of speaker recognition.* Springer, 2011.

[11] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: from features to supervectors," *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[12] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 1937.

[13] S. S. Stevens and J. Volkmann, "The relation of pitch to frequency: A revised scale," *The American Journal of Psychology*, pp. 329–353, 1940.

[14] D. O'shaughnessy, *Speech communication: human and machine*. Universities press, 1987.

[15] S. K. Kopparapu and M. Laxminarayana, "Choice of mel filter bank in computing mfcc of a resampled speech," in *Information Sciences Signal Processing and their Applications (ISSPA), 2010 10th International Conference on*. IEEE, 2010, pp. 121–124.

[16] S. Sigurdsson, K. B. Petersen, and T. Lehn-Schiøler, "Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music," in *Seventh International Conference on Music Information Retrieval (ISMIR)*, Victoria, Canada, 2006.

[17] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1, pp. 19–41, 2000.

[18] R. C. Rose and D. A. Reynolds, "Text independent speaker identification using automatic acoustic segmentation," in *Acoustics, Speech, and Signal Processing, 1990. ICASSP-90., 1990 International Conference on*. IEEE, 1990, pp. 293–296.

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

[20] ——, "Automatic speaker recognition using gaussian mixture speaker models," Citeseer, 1995.

[21] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 1–38, 1977.

[22] D. Reynolds, "An overview of automatic speaker recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)(S. 4072-4075)*, 2002.

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[24] A. Ben-Hur and J. Weston, "A user's guide to support vector machines," in *Data mining techniques for the life sciences*. Springer, 2010, pp. 223–239.

[25] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using gmm supervectors for speaker verification," *Signal Processing Letters, IEEE*, vol. 13, no. 5, pp. 308–311, 2006.

[26] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[27] M. Bahari, "Automatic speaker characterization: Automatic identification of gender, age, language and accent from speech signals," Ph.D. dissertation, Ph. D. dissertation, KU Leuven–Faculty of Engineering Science, Belgium, 2014.

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

[29] ——, "Algorithms for non-negative matrix factorization," pp. 556–562, 2001.

[30] A. T. Cemgil, "Bayesian inference for nonnegative matrix factorisation models," *Computational Intelligence and Neuroscience*, vol. 2009, 2009.

[31] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.

[32] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis," *Neural computation*, vol. 21, no. 3, pp. 793–830, 2009.

[33] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *The Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[34] C. Févotte and A. T. Cemgil, "Nonnegative matrix factorizations as probabilistic inference in composite models," in *Proc. 17th European Signal Processing Conference (EUSIPCO'09)*. Glasgow, Scotland: Citeseer, 2009, pp. 1913–1917.

[35] J. Canny, "Gap: a factor model for discrete data," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*. Sheffield, United Kingdom: ACM, 2004, pp. 122–129.

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

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

[38] J. Paisley, D. Blei, and M. I. Jordan, "Bayesian nonnegative matrix factorization with stochastic variational inference," *Handbook of Mixed Membership Models and Their Applications. Chapman and Hall/CRC*, 2015.

[39] J. Bernardo, M. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, M. West *et al.*, "The variational bayesian em algorithm for incomplete data: with application to scoring graphical model structures," *Bayesian statistics*, vol. 7, pp. 453–464, 2003.

[40] A. Cichocki, R. Zdunek, A. H. Phan, and S.-i. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[41] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 3, pp. 1066–1074, 2007.

[42] P. Sajda, S. Du, and L. C. Parra, "Recovery of constituent spectra using non-negative matrix factorization," in *Optical Science and Technology, SPIE's 48th Annual Meeting*. San Diego, California, United States of America: International Society for Optics and Photonics, 2003, pp. 321–331.

[43] B. Wang and M. D. Plumbley, "Musical audio stream separation by non-negative matrix factorization," in *Proc. DMRN summer conferene*, Glasgow,Scotland, 2005, pp. 23–24.

[44] R. M. Parry and I. Essa, "Estimating the spatial position of spectral components in audio," in *Independent Component Analysis and Blind Signal Separation*. Springer, 2006, pp. 666–673.

[45] D. FitzGerald, M. Cranitch, and E. Coyle, "Non-negative tensor factorisation for sound source separation," 2005.

[46] C. Févotte and A. Ozerov, "Notes on nonnegative tensor factorization of the spectrogram for audio source separation: statistical insights and towards self-clustering of the spatial cues," in *Exploring Music Contents*. Springer, 2011, pp. 102–115.

[47] C. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 320–327, 1976.

[48] K. Sakhnov, E. Verteletskaya, and B. Simak, "Approach for energy-based voice detector with adaptive scaling factor," *IAENG International Journal of Computer Science*, vol. 36, no. 4, p. 394, 2009.

[49] S. G. Tanyer and H. Ozer, "Voice activity detection in nonstationary noise," *IEEE Transactions on speech and audio processing*, vol. 8, no. 4, pp. 478–482, 2000.

[50] S. Mirzaei, H. Van Hamme, and Y. Norouzi, "Bayesian non-parametric matrix factorization for discovering words in spoken utterances," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

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

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

[53] C. M. Bishop, "Bayesian pca," *Advances in neural information processing systems*, pp. 382–388, 1999.

[54] V. Y. Tan and C. Févotte, "Supplementary material to "automatic relevance determination in nonnegative matrix factorization with the β-divergence"," 2012.

[55] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[56] "`fitcsvm` — train binary support vector machine classifier," http://nl.mathworks.com/help/stats/fitcsvm.html, accessed: 2015-05-17.

[57] "`predict` — predict labels for support vector machine classifiers," http://nl.mathworks.com/help/stats/compactclassificationsvm.predict.html, accessed: 2015-05-17.

[58] "`sgolayfilt` — savitzky-golay filtering," http://nl.mathworks.com/help/signal/ref/sgolayfilt.html, accessed: 2015-05-21.

[59] P. Gans and J. B. Gill, "Examination of the convolution method for numerical smoothing and differentiation of spectroscopic data in theory and in practice," *Applied Spectroscopy*, vol. 37, no. 6, pp. 515–520, 1983.

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

[61] H. Christensen, J. Barker, N. Ma, and P. D. Green, "The chime corpus: a resource and a challenge for computational hearing in multisource environments." in *INTERSPEECH*. Citeseer, 2010, pp. 1918–1921.

[62] M. Cooke, J. Barker, S. Cunningham, and X. Shao, "An audio-visual corpus for speech perception and automatic speech recognition," *The Journal of the Acoustical Society of America*, vol. 120, no. 5, pp. 2421–2424, 2006.

[63] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Audio Engineering Society Convention 108*. Audio Engineering Society, 2000.

[64] G.-B. Stan, J.-J. Embrechts, and D. Archambeau, "Comparison of different impulse response measurement techniques," *Journal of the Audio Engineering Society*, vol. 50, no. 4, pp. 249–262, 2002.

[65] D. Campbell, K. Palomaki, and G. Brown, "A matlab simulation of "shoebox" room acoustics for use in research and teaching," *Computing and Information Systems*, vol. 9, no. 3, p. 48, 2005.

[66] D. Campbell, "The roomsim package," June 2007, content available at http://media.paisley.ac.uk/~campbell/Roomsim.

[67] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.