Bar-Ilan University

Multiple Speakers Tracking and Separation using Statistical Inference Methods

Koby Weisberg

Submitted in partial fulfillment of the requirements for the

Master's Degree in the Faculty of Engineering, Bar-Ilan University

Ramat Gan, Israel 2020

This work was carried out under the supervision of Prof. Sharon Gannot,

Faculty of Engineering, Bar-Ilan University.

Contents

Abbreviations

MMSE minimum mean square error

ATF acoustic transfer function

RTF relative transfer function

MVDR minimum variance distortionless response

STFT short-time Fourier transform

RIR room impulse response

PSD power spectral density

RMSE root mean square error

MSE mean square error

SNR signal to noise ratio

BF beamformer

p.d.f. probability density function

ML maximum likelihood

MLE maximum likelihood estimator

EM expectation-maximization

CREM Cappé and Moulines recursive expectation-maximization (EM)

TREM Titterington recursive EM

SRP steered response power

PHAT phase transform

MRF Markov random field

DOA direction of arrival

REM recursive EM

TF time-frequency

TDOA time difference of arrival

GLRT generalized likelihood ratio test

LRT likelihood ratio test

MoG Mixture of Gaussians

MUSIC multiple signals classification

IPD interaural phase difference

LOCATA acoustic source localization and tracking

HMM hidden Markov model

ROC receiver operating characteristic

AUC area under the curve

MAP maximum a posteriori

ULA uniform linear array

SDR source to distortion ratio

SIR source to interference ratio

SAR source to artifacts ratio

LBP loopy belief propagation

ICA independent component analysis

NMF nonnegative matrix factorization

BIU Bar-Ilan university

BP belief propagation

WDO W-disjoint orthogonality

MC Monte-Carlo

FB Forward-Backward

FG Factor Graph

Abstract

In this work, we handle two common tasks of speech processing - DOA tracking, and speech separation - using microphone array.

Working in the short-time Fourier transform (STFT) domain and following the sparsity assumption, only single speaker is active at each time-frequency (TF) bin. We also assume that the DOA is discrete, hence, the speakers' DOA can be one of a predefined set of candidate DOAs. The problem is then formulated as a statistical inference problem, where we aim to infer from the time and frequency observations on both the speakers' DOA, and on the active speaker at each TF bin. The association of each TF bin to a particular speaker, can be used in turn to build a per speaker TF mask, and to separate the STFT signal to the different speakers.

We first determine a statistical model for the microphone array observations given the speech signal, the DOAs and the associations of each TF bin to a speaker. Using the maximum likelihood estimator (MLE) we estimate the speech signal, and following several mathematical manipulations on the conditional probability we show that it can be replaced with the minimum variance distortionless response (MVDR)-beamformer (BF) outputs, applied on each of the candidate DOAs. We then propose three different statistical models for the DOAs and the associations, for each we derive its own inference algorithm which finds those unknowns given the observations. The first is based on Mixture of Gaussians (MoG) model, and we use two variations of the EM algorithm for inference, Batch-EM and Recursive-EM for a static and dynamic case, respectively. The second is either HMM or a variation called Coupled HMM, where for inference we use standard or extended Forward-Backward (FB) algorithm, respectively. The last is a general Factor

Graph (FG) model, where the DOAs are modeled as a Markov chain, and the speaker associations are modeled either independently or as a Markov random field (MRF). For this model we derive a novel inference scheme based on the LBP algorithm.

A comprehensive experimental study demonstrates the benefits of the proposed algorithms in both simulated data and real-life measurements, compared to reference methods.

Chapter 1

Introduction

Multiple-speaker separation is a well-known problem in the speech processing community, aiming to separate the measured microphone signal to its different sources. Another problem of substantial interest is tracking of a moving speaker, which can be used for separation tasks, and is also required in other applications, including navigation, target acquisition and beamforming. Both problems become challenging when multiple moving speakers are concurrently active, as well as when additive interference signals are also captured by the microphone array.

1.1 Literature Survey

Among the most common DOA estimation methods are the steered response power (SRP)-phase transform (PHAT) algorithm [1] and the multiple signals classification (MUSIC) algorithm [2]. However, these techniques are not optimal in the multiple-speaker case, and do not address dynamic scenarios where the sources are moving during the recording. For the separation task, existing algorithms can be roughly divided into four groups: independent component analysis (ICA) algorithms that assume independence of the original source signals [3]; beamforming methods based on the spatial diversity of the speakers; algorithms based on nonnegative matrix factorization (NMF) of the speech PSD; and methods that rely on the sparsity of speech signals in the TF domain [4]. In the latter, the main assumption is that each TF bin is dominated by a single active speaker.

These algorithms usually estimate a separation mask that assigns each TF bin to the active speaker, and use it for separation by applying a mask to the PSD of the measured signal. Comprehensive surveys of separation methods can be found in [5; 6; 7].

Several algorithms address the problem of localization and separation. In [8], the EM algorithm is implemented for estimating both the DOAs and the separation masks of multiple static speakers with a single microphone pair. The algorithm is based on a MoG model defining a grid of possible DOA candidates. Assuming a single dominant speaker in each TF bin, the interaural phase differences (IPDs) from all TF bins are clustered into groups associated with a particular speaker from a candidate DOA. The E-step in the proposed EM iterations provides a soft assignment of each observation to both speaker and DOA. By marginalizing over the DOAs, a separation mask is obtained. The weights of the Gaussians, obtained by the M-step, define a probability distribution on the candidate DOAs, and the DOAs of the active speakers are estimated from the candidates with the highest probabilities. In [9], the algorithm was extended using a MRF model to promote smoothness of the separation mask in both time and frequency, which was shown to improve the separation results. In [10], a dynamic scenario was addressed by two recursive EM (REM) variants, applied to a multichannel extension of the model in [8]: one based on Titterington recursive EM (TREM) [11] and the second based on Cappé and Moulines recursive EM (CREM) [12]. The separation task was not addressed in this paper.

In [13], a multichannel source separation and tracking algorithm was proposed. In this paper, the basic model assumes static sources, and the tracking is applied as a post processing step following the static localization procedure. Here also, the IPDs are used as feature vectors, and are modeled using wrapped distributions. The DOA of each source is computed using circular linear regression, which in the multiple-speaker case, is solved by the EM algorithm. Similar to [8], the E-step is used for estimating the separation mask, and the slopes of the IPDs are transformed to DOAs using the prior knowledge on the inter-channel delay. A dynamic scenario is addressed by first finding the DOAs for each time-step, and then using the estimated DOAs as observations for a factorial wrapped

Kalman filter.

The above papers use the IPD features for the localization task, however with these features, the presence of additive measurement noise is not directly addressed. In [14; 15; 16], the phase-related feature vectors were substituted by the raw STFT observations. In addition, the noise (or reverberation) was explicitly modeled, resulting in improved performance in noisy (or reverberant) scenarios. The observations at the microphone array were modelled as a mixture of multivariate complex-Gaussians with zero-mean, and a spatial covariance matrix consisting of both the speech and the noise PSDs. Furthermore, it was shown in [15] that the PSDs of the candidate speakers can be estimated in advance (prior to the application of the EM algorithm) from the outputs of a set of MVDR-BFs.

The above algorithms do not provide an explicit DOA estimate, but rather a probability map over the candidate DOAs. While for the static localization task the actual DOA can be found relatively easily by finding the peaks in the probability map, in a dynamic case the peaks should be calculated for each time-step rendering the explicit trajectory inference difficult.

Another approach to address the tracking task is to substitute the MoG model with an HMM. In this approach, the DOAs of the speakers are also discretized to a finite set of candidates. The model assumes that the dynamics of the sources is governed by a Markov process, with higher probability for switching from one candidate to an adjacent candidate at each time-step, thus allowing small changes in the DOA [17; 18].

The tasks of tracking and separation depend on each other. The reason is that when the DOAs of the speakers are known, we can identify the dominating DOA in each TF bin and associate it with the corresponding speaker, and thus extract it by masking. In the opposite direction, given the association map that relates each TF bin to its dominating speaker, we can use the set of TF bins attached with each speaker to infer its corresponding DOA. Examples of using the outcomes of localization to perform separation can be found in [8; 10; 15; 19], and for the other direction in [20; 21].

A simultaneous tracking and separation algorithm was proposed in [18] using a

Bayesian approach. The definition of the hidden variables here is different from that defined in [8]. In [8] each TF observation is associated with both a DOA and a speaker, whereas in [18] each observation is associated only with a speaker, and the speaker is associated with a DOA. This approach uses fewer hidden variables, hence reducing the computational requirements, while modeling real scenarios more accurately. The continuous movement of the speakers is reflected by modelling the DOAs of the speakers as Markov processes. Since an exact inference of the hidden variables from the observations is intractable, a variational inference was applied.

1.2 Main Contribution

In the current contribution we formulate these problems as a statistical inference problems, where the hidden data are either the DOA of each TF bin (as in [15]), or both the association of each bin to the active speaker in this bin and the DOA of each speaker at each time-stamp (as in [18]). In order to estimate the hidden data given the observations, one needs to define statistical model for both the hidden data, and for the observations given the hidden data. For the latter we use a model similar to [15], and we show that the raw observation features can be substituted by new features, which are the likelihood ratio test (LRT) at each candidate DOA indicating whether the MVDR-BF output at this DOA dominated by either speech or noise. The utilization of these new features, results in a lower computational burden that is beneficial in online and real-time applications. For the hidden data, three different models are proposed, based on our papers [22; 23; 24], described in the following.

The first, presented in Chapter 3, is the instantaneous model, where the hidden variables are the DOA of each TF bin, which assumed to be statistically independent with shared prior probability. In this model the marginal distribution of the observations is MoG and the inference is done using the EM algorithm. We further propose a tracking procedure for dynamic scenario by applying the CREM algorithm.

In the second model, presented in Chapter 4, the hidden data is similarly defined, however, it is modeled using two variants of HMM. In the first, a frequency-dependent HMM with the DOAs of the observations as the hidden Markov process, is used to obtain a smooth track of the sources. Since the set of active frequencies can vary across time-frames, we extend the model by introducing the coupled HMM paradigm [25]. In both models the emission probabilities of the HMMs are the LRT outputs. The inference procedure is then implemented by an extended FB algorithm [26]. The results of this procedure is a smooth DOA posterior probability per TF bin. Finally, a per-frame probability map of the DOAs is obtained by frequency averaging.

The above algorithms provide a per-frame probability map of the speakers DOAs, and any peak-picking method can then be applied to this map to extract a time-varying DOA estimate for each speaker. In Chapter 5 we present a factor graph model, where in contrast to the two previous models, the hidden variables are both the speakers DOA and the TF bins association to speakers. By estimating those variables, both separation mask and explicit DOA trajectory is obtained for each speaker.

Factor graph models [27] are used in many complex tasks in various signal processing fields, such as communication [28], sonar detection [29] and robotics [30; 31]. To the best of our knowledge, this model was not used for the task of speaker tracking and separation. In the factor graph model, we define the hidden data as in [18] using two groups of latent variables. The first group consists of the DOA of the sources that are modeled as separated Markov chain for each source, where the transition probability is set to allow only small changes in the DOAs in subsequent time steps. The second group consists of the associations of the TF bins to the different sources, which can be modeled by an i.i.d. distribution or, following [9], using a MRF model to smoothen the associations in time and frequency. We then show that the posterior of the latent variables given the observations defines a factor graph, and we derive a novel inference method for simultaneously estimating all latent variables, using the loopy belief propagation (LBP) inference algorithm [32].

The algorithms proposed in this work are summarized in table 1.1.

Model name	Graphical model	Inference algorithm	Chapter	Where published
Instantaneous	None	EM/Recursive EM	3	[22]
Parrallel HMM	HMM	FB	4.1	[23]
Coupled HMM	Coupled HMM	Extended FB	4.2	[23]
Factor graph	Factor Graph	LBP	5	[24]

Table 1.1: Summary of the algorithms proposed in this work.

Chapter 2

Problem Statement and Observations Model

In this chapter we will formulate the tracking and separation tasks as a statistical inference problem, where we aim to infer from the observations on the unobserved ("hidden") data. In addition, we will define the statistical model of the observations given the hidden data, which is identical for all of our proposed algorithms, described later in this work. The difference between our proposed methods lies in the statistical model of the hidden data, as will be explained in details in Chapters 3, 4 and 5.

2.1 Problem Formulation

Consider an array of N microphones, receiving signals of J moving speakers. At each time step, each speaker is located at a specific DOA on a grid of M possible DOAs $[\vartheta_1, \ldots, \vartheta_M]$. Due to the dynamic nature of the problem, the DOAs may vary from one time step to the other. The proposed method is applied in the STFT domain with $t = 1, \ldots, T$ denoting the time index and $k = 1, \ldots, K$ denoting the frequency index. Let $d_t(j)$ be a categorical random variable denoting the DOA index of the jth speaker at time index t, i.e. $d_t(j) \in [1, \ldots, M]$. Relying on the W-disjoint orthogonality (WDO) property of speech signals in the STFT domain [4], it can be assumed that each TF bin is dominated by a single active speaker. Let $a_{t,k}$ be a categorical random variable denoting the active speaker at the (t, k)th bin, i.e. $a_{t,k} \in [1, \dots, J]$. Following these definitions, the nth microphone signal is given by:

$$z_{t,k}^{(n)} = g_k^{(n)}(d_t(a_{t,k}))s_{t,k}(d_t(a_{t,k})) + v_{t,k}^{(n)}, \tag{2.1}$$

where $d_t(a_{t,k}) \in [1, \dots, M]$ is the DOA index of the active speaker at the (t, k)th bin, $g_k^{(n)}(m)$ is the relative transfer function (RTF) associated with the mth candidate DOA and defined between the nth microphone and the reference microphone, $s_{t,k}(m)$ is the speech signal from the mth candidate as measured by the reference microphone, and $v_{t,k}^{(n)}$ denotes a stationary ambient noise at microphone $n \in [1, \dots, N]$.

In low-reverberation environments, the RTF approximately corresponds to the direct path between the source and the microphone:

$$g_k^{(n)}(m) = \exp\left(-\iota \frac{2\pi k}{K} \frac{\tau_{m,n}}{T_s}\right) \tag{2.2}$$

where T_s denotes the sampling period, and $\tau_{m,n}$ denotes the known time difference of arrival (TDOA) between the nth microphone and the reference microphone, associated with the mth candidate DOA.

The measured signals (2.1) can be written in a vector form as:

$$\mathbf{z}_{t,k} = \mathbf{g}_k(d_t(a_{t,k}))s_{t,k}(d_t(a_{t,k})) + \mathbf{v}_{t,k} \tag{2.3}$$

where

$$\begin{aligned}
\mathbf{z}_{t,k} &= \left[z_{t,k}^{(1)}, z_{t,k}^{(2)}, \dots z_{t,k}^{(N)}\right]^T \\
\mathbf{g}_k(m) &= \left[1, g_k^{(2)}(m), \dots, g_k^{(N)}(m)\right]^T \\
\mathbf{v}_{t,k} &= \left[v_{t,k}^{(1)}, v_{t,k}^{(2)}, \dots, v_{t,k}^{(N)}\right]^T .
\end{aligned}$$

assuming, without loss of generality, that the first microphone is chosen as the reference microphone. The generation of the observation by the defined model is illustrated in Figure 2.1.

In the following we will denote $\mathbf{a} = \text{vec}_{t,k}\{a_{t,k}\}$ and $\mathbf{d} = \text{vec}_{t,j}\{d_t(j)\}$ as the hidden data, and $\mathbf{z} = \text{vec}_{t,k}\{\mathbf{z}_{t,k}\}$, as the observations. In some cases, it is more convenient to define the DOA association of each TF bin, namely $b_{t,k} \equiv d_t(a_{t,k}) \in [1, \ldots, M]$ as the hidden data, and we will define accordingly $\mathbf{b} = \text{vec}_{t,k}\{b_{t,k}\}$. Our goal is to estimate the hidden data given the observations. To this end, we need to define a statistical model and to present an inference scheme that estimates the hidden data.

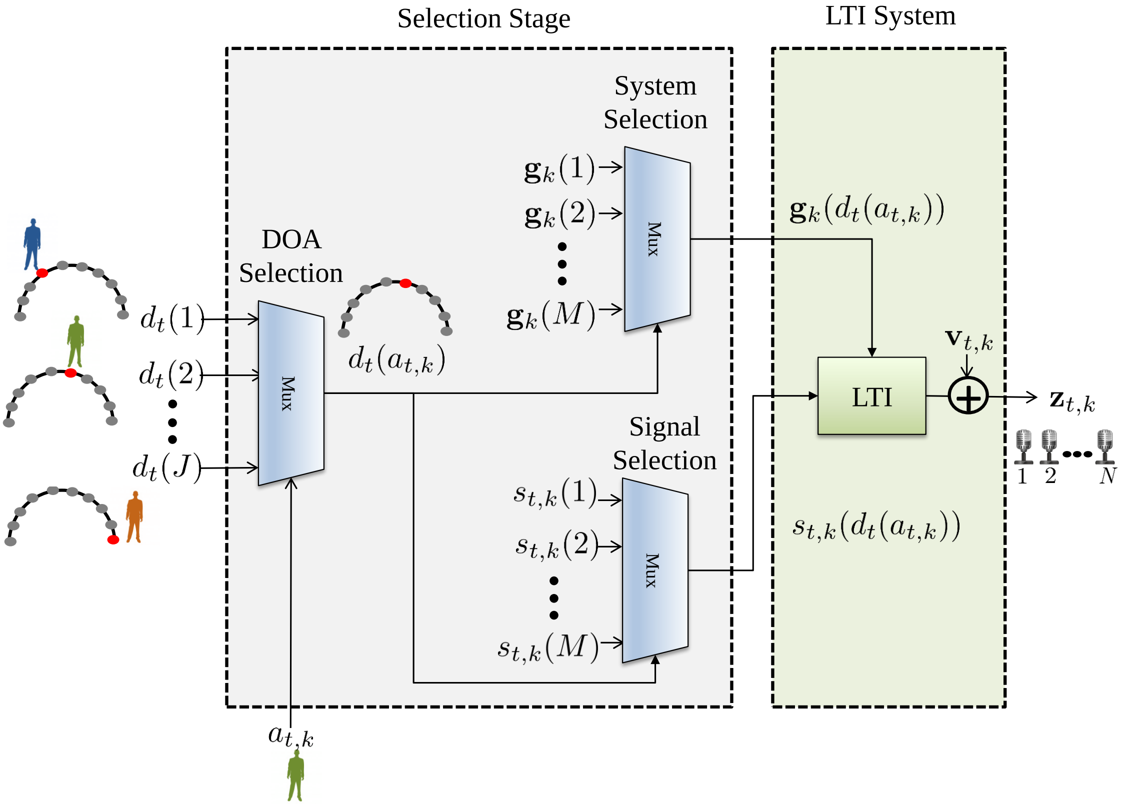


Figure 2.1: An illustration of the generation of the observations by the presented model. The first part is the selection stage. The variable $a_{t,k}$ representing the active speaker, is used for selecting the DOA associated with the active speaker. The chosen DOA candidate is used for selecting both the RTF and the input speech signal that are associated with this candidate. The second part describes the actual generation of the observations by an LTI system model, in which the chosen speech signal is filtered by the chosen RTF and noise is added.

2.2 The observations model

The statistical model of the hidden variables, either $P(\mathbf{a}, \mathbf{d})$ or $P(\mathbf{b})$ will be discussed latter in Chapters 3, 4 and 5. We will now define the statistical model of the observations given the hidden variables $P(\mathbf{z}|\mathbf{a}, \mathbf{d})$. The speech signal is modeled as a zero-mean complex-Gaussian random variable with a time-varying PSD:

$$P(s_{t,k}(d_t(a_{t,k}))) = \mathcal{N}\left(s_{t,k}(d_t(a_{t,k})); 0, \phi_{s,t,k}(d_t(a_{t,k}))\right) \tag{2.4}$$

where $\mathcal{N}(\cdot; \cdot, \cdot)$ denotes the complex-Gaussian probability and $\phi_{s,t,k}(d_t(a_{t,k}))$ is the unknown PSD of the speech signal received from the DOA of the active speaker at the (t, k)th bin. The noise is modeled as a zero-mean complex-Gaussian random vector with a time-invariant covariance matrix $\Phi_{\mathbf{v},k}$:

$$P(\mathbf{v}_{t,k}) = \mathcal{N}\left(\mathbf{v}_{t,k}; \mathbf{0}, \Phi_{\mathbf{v},k}\right). \tag{2.5}$$

It is assumed that the noise covariance matrix is known in advance, or can be estimated during speech-absent segments, due to the noise stationarity.

Following equations (2.3), (2.4) and (2.5), the conditional probability density function (p.d.f.) of the (t, k)th observation given the DOA of the active speaker at this bin can be expressed as

$$P(\mathbf{z}_{t,k}|d_t(a_{t,k})) = \mathcal{N}(\mathbf{z}_{t,k}, \mathbf{0}, \Phi_{\mathbf{z},t,k}(d_t(a_{t,k}))), \tag{2.6}$$

with:

$$\Phi_{\mathbf{z},t,k}(m) = \mathbf{g}_k(m)\mathbf{g}_k^H(m)\phi_{s,t,k}(m) + \Phi_{\mathbf{v},k}, \tag{2.7}$$

where the speech and noise signals are assumed to be statistically independent.

2.3 Likelihood simplification

We will now simplify the conditional probability of the observations given the hidden variables (2.6). We first factorize the probability, then we estimate the speech PSD using the maximum likelihood estimator (MLE) and finally we substitute the estimated PSD in the factorized probability to obtain the final simple expression.

2.3.1 Likelihood factorization

We factorize the likelihood of the observation to obtain a simpler expression. We first define the a priori signal to noise ratio (SNR) of the signal impinging the array from the mth candidate position as:

$$\zeta_{t,k}(m;\phi_{s,t,k}(m)) = \frac{\phi_{s,t,k}(m)}{\phi_{v,k}(m)}. \tag{2.8}$$

and the a posteriori SNR as:

$$\eta_{t,k}(m) = \frac{|\hat{s}_{\mathbf{w},t,k}(m)|^2}{\phi_{v,k}(m)} \tag{2.9}$$

where $\hat{s}_{\mathbf{w},t,k}(m)$ is the output of an MVDR-BF directed towards the mth candidate:

$$\hat{s}_{\mathbf{w},t,k}(m) \equiv \mathbf{w}_k^H(m)\mathbf{z}_{t,k} \tag{2.10}$$

where the MVDR-BF is defined by:

$$\mathbf{w}_k(m) = \frac{\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)}{\mathbf{g}_k^H(m)\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)}, \tag{2.11}$$

and $\phi_{v,k}(m)$ is the PSD of the residual noise at the output of the MVDR-BF, and is given by:

$$\phi_{v,k}(m) \equiv \frac{1}{\mathbf{g}_k^H(m)\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)}. \tag{2.12}$$

According to (2.6), the conditional distribution of a single observation given the hidden data is given by:

$$\mathcal{N}(\mathbf{z}_{t,k}, \mathbf{0}, \Phi_{\mathbf{z},t,k}(m)) = \frac{1}{\pi^N \det(\Phi_{\mathbf{z},t,k}(m))} \exp(-\mathbf{z}^H \left(\Phi_{\mathbf{z},t,k}(m)\right)^{-1} \mathbf{z}). \tag{2.13}$$

Using the definition of $\Phi_{\mathbf{z},t,k}(m)$ (2.7) and Sylvester's determinant theorem, the determinant can be written as:

$$\begin{aligned}\det(\Phi_{\mathbf{z},t,k}(m)) &= \det(\Phi_{\mathbf{v},k}) \cdot \det(1 + \phi_{s,t,k}(m)\mathbf{g}_k^H(m)\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)) \\ &= \det(\Phi_{\mathbf{v},k}) \cdot (1 + \zeta_{t,k}(m; \phi_{s,t,k}(m))).\end{aligned}$$

In addition, using the Woodbury identity, the inversion of $\Phi_{\mathbf{z},t,k}(m)$ can be written as:

$$\Phi_{\mathbf{z},t,k}(m)^{-1} = \Phi_{\mathbf{v},k}^{-1} - \frac{\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)\mathbf{g}_k^H(m)\Phi_{\mathbf{v},k}^{-1}}{\phi_{s,t,k}(m)^{-1} + \mathbf{g}_k^H(m)\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)}. \tag{2.14}$$

By substituting these relations into the p.d.f., we can factorize it as following:

$$\mathcal{N}(\mathbf{z}_{t,k}, \mathbf{0}, \Phi_{\mathbf{z},t,k}(m)) = T_{t,k}(m; \phi_{s,t,k}(m)) \cdot G_{t,k} \tag{2.15}$$

where $G_{t,k}$ aggregates all terms which do not depend on m:

$$G_{t,k} = \frac{1}{\pi^N \det(\Phi_{\mathbf{v},k})} \exp\left(-\mathbf{z}^H \Phi_{\mathbf{v},k}^{-1}\mathbf{z}\right) \equiv \mathcal{N}(\mathbf{z}_{t,k}, \mathbf{0}, \Phi_{\mathbf{v},k}) \tag{2.16}$$

and $T_{t,k}(m; \phi_{s,t,k}(m))$ aggregates the other terms:

$$\begin{aligned}T_{t,k}(m; \phi_{s,t,k}(m)) = &\frac{1}{1 + \zeta_{t,k}(m; \phi_{s,t,k}(m))} \\ &\cdot \exp\left(\frac{\mathbf{z}^H\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)\mathbf{g}_k^H(m)\Phi_{\mathbf{v},k}^{-1}\mathbf{z}}{\phi_{s,t,k}(m)^{-1} + \mathbf{g}_k^H(m)\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)}\right).\end{aligned}$$

Using (2.10),(2.12), (2.9) and (2.8) we can write $T_{t,k}(m;\phi_{s,t,k}(m))$ in a simple way:

$$T_{t,k}(m;\phi_{s,t,k}(m)) = \frac{1}{1+\zeta_{t,k}(m;\phi_{s,t,k}(m))} \exp\left(\frac{\zeta_{t,k}(m;\phi_{s,t,k}(m))\eta_{t,k}(m)}{1+\zeta_{t,k}(m;\phi_{s,t,k}(m))}\right). \quad (2.17)$$

Note that $T_{t,k}(m;\phi_{s,t,k}(m))$ is the likelihood ratio test (LRT), as presented in [33, Eq. (14)]. The LRT tests whether $\mathbf{z}_{t,k}$ is either associated with a speaker located in the mth candidate DOA or with noise only. The computation of $T_{t,k}(m)$ is described in Algorithm 1.

Finally we obtain for the conditional probability for each TF bin observation:

$$P(\mathbf{z}_{t,k}|d_t(a_{t,k})) = T_{t,k}(d_t(a_{t,k})) \cdot G_{t,k} \quad (2.18)$$

and assuming independence between the different TF bins observations given the latent variables, the *likelihood* of the entire set of the observations is given by:

$$P(\mathbf{z}|\mathbf{a},\mathbf{d}) = \prod_{t,k} T_{t,k}(d_t(a_{t,k})) \cdot G_{t,k}. \quad (2.19)$$

2.3.2 Speech PSD estimation

In this section we substitute the hidden variables $a_{t,k}$ and $d_t(j)$ with $b_{t,k} = d_t(a_{t,k})$ for simplicity. Since $\phi_{s,t,k}(m)$ does not directly depend on the identity of the active speaker but on its DOA, we can estimate it prior to the algorithm application using the maximum likelihood estimator (MLE). To this end, we write the marginal distribution of the observations, by marginalizing out the hidden variables:

$$P(\mathbf{z};\boldsymbol{\phi_s}) = \sum_{\mathbf{b}} \prod_{t,k} P(\mathbf{z}_{t,k}|\mathbf{b}_{t,k})P(\mathbf{b}) \quad (2.20)$$

where $P(\mathbf{b})$ is the prior probability of $\mathbf{b}$ which depends on the priors $P(\mathbf{a})$ and $P(\mathbf{d})$.

The MLE for $\phi_{s,\tilde{t},\tilde{k}}(m)$ is obtained by maximizing (2.20) w.r.t. $\phi_{s,\tilde{t},\tilde{k}}(m)$. We first rearrange the marginal distribution by excluding the $(\tilde{t},\tilde{k})$th observation from the product and summation:

$$P(\mathbf{z};\boldsymbol{\phi_s}) = \sum_{\mathbf{b}_{\tilde{t},\tilde{k}}} \left[P(\mathbf{z}_{\tilde{t},\tilde{k}}|\mathbf{b}_{\tilde{t},\tilde{k}}) \sum_{\substack{\mathbf{b}\backslash \\ \mathbf{b}_{\tilde{t},\tilde{k}}}} \prod_{\substack{t,k\backslash \\ (\tilde{t},\tilde{k})}} P(\mathbf{z}_{t,k}|\mathbf{b}_{t,k}) P(\mathbf{b}) \right]. \tag{2.21}$$

Substituting (2.6) into (2.21), and explicitly writing the first summation over all candidates, we have:

$$P(\mathbf{z};\boldsymbol{\phi_s}) = \sum_{w=1}^{M} \mathcal{N}(\mathbf{z}_{\tilde{t},\tilde{k}}, \mathbf{0}, \Phi_{\mathbf{z},\tilde{t},\tilde{k}}(w)) \cdot C \tag{2.22}$$

where $C \equiv \sum_{\mathbf{b}\backslash\mathbf{b}_{\tilde{t},\tilde{k}}} \prod_{t,k\backslash(\tilde{t},\tilde{k})} P(\mathbf{z}_{t,k}|\mathbf{b}_{t,k})P(\mathbf{b})$ denotes a positive term, independent of the parameter of interest $\phi_{s,\tilde{t},\tilde{k}}(m)$. Then, taking the derivative w.r.t $\phi_{s,\tilde{t},\tilde{k}}(m)$ we get:

$$\frac{\partial P(\mathbf{z};\boldsymbol{\phi_s})}{\partial \phi_{s,\tilde{t},\tilde{k}}(m)} = \frac{\partial \mathcal{N}(\mathbf{z}_{t,k}, \mathbf{0}, \Phi_{\mathbf{z},t,k}(m))}{\partial \phi_{s,\tilde{t},\tilde{k}}(m)} \cdot C. \tag{2.23}$$

By setting this derivative to zero we get the MLE for $\phi_{s,t,k}(m)$ [34]:

$$\hat{\phi}_{s,t,k}(m) = |\hat{s}_{\mathbf{w},t,k}(m)|^2 - \phi_{v,k}(m). \tag{2.24}$$

where $\hat{s}_{\mathbf{w},t,k}(m)$ is the MVDR-BF output defined in (2.10), and $\phi_{v,k}(m)$ is the PSD of the residual noise at the output of the MVDR-BF defined in (2.12).

Using the estimator of $\phi_{s,t,k}(m)$ we can further simplify $T_{t,k}(m;\phi_{s,t,k}(m))$. Dividing (2.24) by $\phi_{v,k}(m)$ and using the definitions in (2.9) and (2.8), we obtain:

$$\zeta_{t,k}(m;\hat{\phi}_{s,t,k}(m)) = \eta_{t,k}(m) - 1. \tag{2.25}$$

By substituting this relation into (2.17), we finally obtain:

$$T_{t,k}(m) = T_{t,k}(m;\hat{\phi}_{s,t,k}(m)) = \frac{1}{\eta_{t,k}(m)} \exp\left(\eta_{t,k}(m) - 1\right). \tag{2.26}$$

Algorithm 1 Likelihood calculation

```
• Calculate the MVDR-BF $\mathbf{w}_k(m)\ \forall k, m$ using (2.11)
• Calculate the output of the MVDR-BF $\hat{s}_{\mathbf{w},t,k}(m)\ \forall t, k, m$ using (2.10)
• Calculate the PSD of the residual noise $\forall k, m$:
      $\phi_{v,k}(m) \equiv \frac{1}{\mathbf{g}_k^H(m)\Phi_{\mathbf{v},k}^{-1}\mathbf{g}_k(m)}$
• Calculate the SNR at the output of the MVDR-BF $\forall t, k, m$:
      $\eta_{t,k}(m) = \frac{|\hat{s}_{\mathbf{w},t,k}(m)|^2}{\phi_{v,k}(m)}$
• Calculate the LRT $\forall t, k, m$:
      $T_{t,k}(m; \hat{\phi}_{s,t,k}(m)) = \frac{1}{\eta_{t,k}(m)} \exp\left(\eta_{t,k}(m) - 1\right)$
```

Chapter 3

Instantaneous Hidden Data Model

The material presented in this chapter is based on [22]:

> K. Weisberg, S. Gannot, and O. Schwartz, "An online multiple-speaker doa tracking using the Cappé-Moulines recursive expectation-maximization algorithm," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 656–660.

In this section we will present the instantaneous hidden data model. The hidden data is defined to be the DOA of the active speaker at each TF bin, while in-dependency is assumed along time and frequency. In this model we aim to solve only the tracking problem, using either batch EM or recursive EM algorithm.

3.1 The hidden data model

In order to simplify the inference procedure, the hidden data is defined as the DOA associations of each TF bin $b_{t,k}$, and it assumed that those variables are independent along the TF bins with:

$$P(b_{t,k} = m) = \psi_m \tag{3.1}$$

where ψ_m is the a priori probability of the activity of a speaker at the mth position, and $\sum_{m=1}^{M} \psi_m = 1$. Because the actual number of speakers is usually lower than the number of candidates, most of ψ_m will be close to zero [35]. A graphical representation of this model is shown in Fig. 3.1 (top). Following this definition the marginal distribution of the observations is MoG, and we can write the p.d.f. of the entire set of observations as:

$$P(\mathbf{z};\boldsymbol{\theta}) = \prod_{t,k} \sum_{m=1}^{M} \psi_m T_{t,k}(m; \phi_{s,t,k}(m)) \cdot G_{t,k} \tag{3.2}$$

where $\boldsymbol{\theta}$ is the set of unknown parameters, namely $\boldsymbol{\theta} = \left[\boldsymbol{\psi}^T, \boldsymbol{\phi_s}^T\right]^T$ with $\boldsymbol{\psi} = \text{vec}_m \left\{\psi_m\right\}$ and $\boldsymbol{\phi_s} = \text{vec}_{t,k,m} \left\{\phi_{s,t,k}(m)\right\}$, and we used the factorized likelihood from 2.15. The maximum likelihood (ML) problem can readily be stated as: $\widehat{\boldsymbol{\theta}} = \text{argmax}_{\boldsymbol{\theta}} \log f(\mathbf{z}; \boldsymbol{\theta})$. Note that although the parameters $\boldsymbol{\phi_s}$ can be estimated in advance, as described in Sec. 2.3.2, we write them here as unknown parameters. This will facilitate the derivation of recursive algorithm, as detailed in Chapter. 3.3.

3.2 Localization using Batch EM

In the batch-EM, we assume that $\phi_{s,t,k}(m)$ is changing independently over time, and therefore can be calculated in advance as derived above Sec. 2.3.2. An alternative approach is to apply the EM algorithm to infer this parameter as in [22], however, both approaches obtain the same estimator.

The auxiliary function of the EM algorithm is given by:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(\ell-1)}) = E\left\{\log\left(P(\mathbf{z},\mathbf{b};\boldsymbol{\theta})\right)|\mathbf{z};\boldsymbol{\theta}^{(\ell-1)}\right\} \tag{3.3}$$

where the joint p.d.f. of the observations and the hidden data (the complete data) is given by:

$$P(\mathbf{z},\mathbf{b};\boldsymbol{\theta}) = P(\mathbf{z}|\mathbf{b})P(\mathbf{b};\boldsymbol{\psi}). \tag{3.4}$$

The E-step is then given by:

$$\hat{b}_{t,k}^{(\ell-1)}(m) = E\left\{b_{t,k}(m)|\mathbf{z}_{t,k}(m);\boldsymbol{\theta}^{(\ell-1)}\right\} == \frac{\psi_m^{(\ell-1)}T_{t,k}(m;\hat{\phi}_{s,t,k}(m))\cdot G_{t,k}}{\sum_m \psi_m^{(\ell-1)}T_{t,k}(m;\hat{\phi}_{s,t,k}(m))\cdot G_{t,k}}, \tag{3.5}$$

and the M-step by:

$$\hat{\psi}_m^{(\ell)} = \frac{\sum_{t,k}\hat{b}_{t,k}^{(\ell-1)}(m)}{T\cdot K}. \tag{3.6}$$

Since $G_{t,k}$ is not depend on m, it can cancelled out in the E-step, and therefore we obtain simpler expression:

$$\hat{b}_{t,k}^{(\ell-1)}(m) = \frac{\psi_m^{(\ell-1)}T_{t,k}(m;\hat{\phi}_{s,t,k}(m))}{\sum_m \psi_m^{(\ell-1)}T_{t,k}(m;\hat{\phi}_{s,t,k}(m))} \tag{3.7}$$

where $T_{t,k}(m;\hat{\phi}_{s,t,k}(m))$ is defined in (2.26).

3.3 Recursive EM

In this section, we will apply the CREM algorithm, presented in [12], to the problem at hand. To allow for a smooth estimate of the speech PSD, we introduce time-dependency between frames, i.e. $\hat{\phi}_{s,t,k}(m)$ depends on a set of frames. The (smooth) time-variations of the speech PSD will be naturally obtained by the recursive nature of the algorithm. In the CREM scheme, the iteration index ℓ is substituted by the time index t, and the recursive auxiliary function is based on smoothing of the instantaneous auxiliary function over time:

$$Q_R(t;\boldsymbol{\theta}) = (1-\gamma)Q_R(t;\boldsymbol{\theta}) + \gamma Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t-1)) \tag{3.8}$$

where $Q_R(t;\boldsymbol{\theta})$ is the recursive auxiliary function, and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}(t-1))$ is the instantaneous auxiliary function given only the current observations. The M-step is obtained by maximizing $Q_R(t;\boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$. Using (3.3) and (3.4) the recursion in (3.8) boils down to:

$$\eta_{t,k}(m) = (1-\gamma)\eta_{t-1,k}(m) + \gamma\hat{b}_{t,k}(m), \tag{3.9a}$$

$$\xi_{t,k}(m) = (1-\gamma)\xi_{t-1,k}(m) + \gamma\hat{b}_{t,k}(m)\left|\hat{s}_{\mathbf{w},t,k}(m)\right|^2. \tag{3.9b}$$

Maximizing $Q_R(t;\boldsymbol{\theta})$ with respect to ψ_m and ϕ_s yields the M-step:

$$\hat{\psi}_t(m) = \frac{\sum_k \eta_{t,k}(m)}{K} \tag{3.10}$$

$$\hat{\phi}_{s,t,k}(m) = \frac{\xi_{t,k}(m)}{\eta_{t,k}(m)} - \phi_{v,k}(m). \tag{3.11}$$

A recursive estimator of $\hat{b}_{t,k}(m)$ can be obtained from the CREM by substituting $\hat{\psi}_m^{(\ell)}$ with $\hat{\psi}_m^{(t)}$ in (3.7) and by using the original LRT expression from (2.17) with the smoothed estimator of $\phi_{s,t,k}(m)$:

$$T_{t,k}(m;\hat{\phi}_{s,t-1,k}(m)) = \frac{1}{1+\zeta_{t,k}(m;\hat{\phi}_{s,t-1,k}(m))}\exp\left(\frac{\zeta_{t,k}(m;\hat{\phi}_{s,t-1,k}(m))\eta_{t,k}(m)}{1+\zeta_{t,k}(m;\phi_{s,t,k}(m))}\right). \tag{3.12}$$

Note the significant differences between (2.26) and (3.12). While the former does not take into account the smoothness of the speech PSD, and hence uses only an instantaneous SNR estimate; the latter takes the smoothness of the PSD into account through the recursively estimated a priori SNR estimate. We also note that the a priori SNR estimate obtained here by the CREM procedure is very different from the estimators presnted in [33]. An illustration of the difference between the batch and the recursive algorithms is presented in 3.2.

3.4 Practical considerations

The original CREM uses one smoothing parameter γ. We note that in our problem, the two parameters exhibit different time behaviors: while $\boldsymbol{\psi}$, which is related to the source position, is slowly time-varying, the speech PSD $\phi_{s,t,k}(m)$ is rapidly changing. Therefore, in our experiments, we used two different smoothing parameters: γ_ψ and γ_{ϕ_s}. Accordingly, for estimating $\xi_{t,k}(m)$, we always used $\gamma_{\phi_s} \approx 1$. For $\eta_{t,k}(m)$, we used two estimators: the first one used $\gamma_{\phi_s} \approx 1$ to obtain an estimate for $\phi_{s,t,k}(m)$ in (3.11), and the second used $\gamma_\psi \ll 1$ to obtain an estimate of $\boldsymbol{\psi}$ in (3.10).

3.5 Experimental study

The proposed algorithm was evaluated using two data-sets: simulated time-varying scenes generated by a signal generator[1] and real multichannel audio recordings from the LOCATA challenge [36].

3.5.1 Algorithm settings and baseline methods

The parameters used in the implementation of our algorithm are as follows: 1) signals re-sampled to 16 kHz; 2) STFT frame-length 64 ms with no overlap; 3) frequency band used for localization $1-6$ KHz; 4) smoothing parameters $\gamma_\psi = 0.1$, $\gamma_{\phi_s} = 0.8$; 5) grid of possible azimuth angle between -90^0 and 90^0, with resolution 2° and 5° for the simulated data and LOCATA data-set, respectively; and 6) the probabilities were uniformly initialized to $\hat{\psi}_t(m) = \frac{1}{M}, \forall m$. The noise PSD matrix was estimated using speech absence segment at the beginning of the recording, annotated manually for the LOCATA data-set.

The proposed method provides a probability map as a function of time and not directly the DOA estimates. For estimating the actual trajectory of the speakers, one should use a peak-selection method. To circumvent the effects of the peak-selection algorithm, we have chosen to calculate instead the receiver operating characteristic (ROC) curve for each frame and to use the area under the curve (AUC) as a measure. For calculating the ROC curve, all detections in the range around the true DOA, specifically $\text{DOA}_{\text{gt}} \pm 3^\circ$, are considered *true positive*. The final score is obtained by time-averaging of the per-frame AUC, excluding noise-only frames. For baseline methods, we used both the MUSIC

[1] www.audiolabs-erlangen.de/fau/professor/habets /software/signal-generator

algorithm [2], as provided by the challenge, and the PRP-REM algorithm [10] with the same smoothing parameter, and with fixed variance for all the Gaussians, $\sigma = 0.1$. For a fair comparison, the MUSIC results were similarly smoothed and normalized to obtain a pseudo-distribution.

3.5.2 Evaluation using simulated data

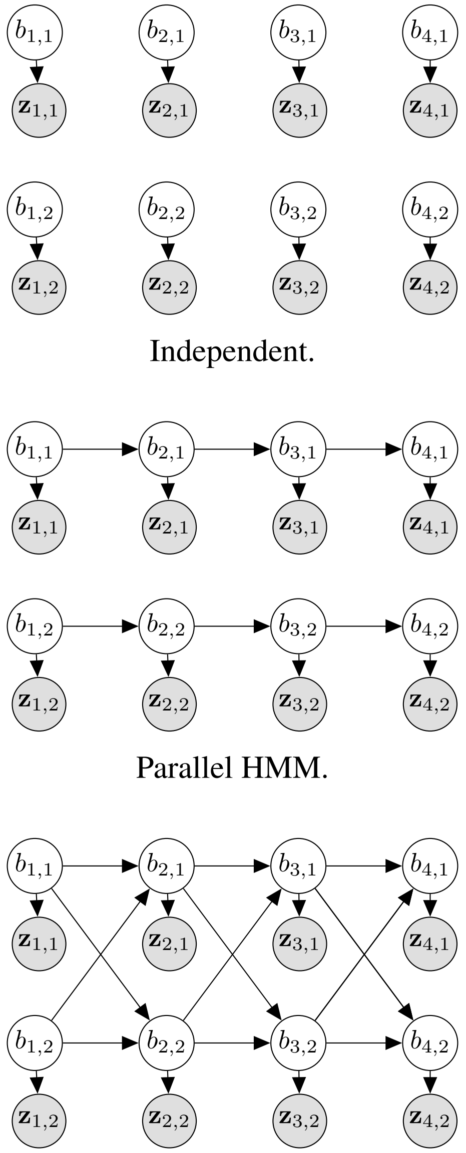
In the simulated scenario, clean anechoic speech signals were drawn from the TIMIT database [37], where speech utterances of the same speaker were concatenated to obtain a 5 s long speech signal. The speakers were randomly selected from 26 different speakers. To simulate moving sources, we used the signal generator, as mentioned above. The room dimensions were set to $6 \times 6 \times 6.1$ m with reverberation time $T_{60} \sim 200$ ms. The signals were captured by an eight-microphone linear array with inter-distances of $[3, 3, 3, 8, 3, 3, 3]$ cm from one another, together with an additive spatially-white noise with various SNR values.

Thirty Monte-Carlo trials, simulating two moving sources scenarios, were examined. In each scenario, the initial DOAs of the speakers were set to $60°$ and $100°$, respectively. The sources moved from their initial positions in a circle with a radius of 1 m around the array center and with angular velocity randomly selected from a uniform distribution in the range $[-15 : 15]$ $\frac{\text{deg}}{\text{s}}$ to obtain random trajectories. We first examined the influence of γ_{ϕ_s} on the obtained localization score. We have noticed that the scores are insensitive to the smoothing parameter value in the range $0.6 < \gamma_{\phi_s} < 0.9$. We have therefore selected $\gamma_{\phi_s} = 0.8$ for all experiments.

The results of the simulation study are depicted in Fig. 3.3. It is evident from Fig. 3.3(a) that the proposed algorithm outperforms the PRP-REM algorithm [10] by approximately 5% for 0 dB SNR, and that their performance converges as the SNR level increases. It is also demonstrated that the proposed method significantly outperforms the MUSIC algorithm. Moreover, we note that the proposed method is computationally more efficient than the PRP-REM, and that it additionally provides the speech PSD estimate that may be useful for further processing, e.g. in separation tasks [15]. In Fig. 3.3(b) we depict the probability map $\hat{\psi}_m$ of one the trials, clearly demonstrating the tracking capabilities of the proposed method.

3.5.3 Evaluation on LOCATA data-set

The data for the LOCATA challenge [36] were recorded in a room of size $7.1 \times 9.8 \times 3$ m with a reverberation time $T60 \sim 0.55s$. We tested our algorithm on Task #3, which is a recording of a single moving speaker, and Task #4, which is a recording of two moving speakers. We used the data recorded by the linear array (DICIT). We used the first recording (Recording #1) of each task. As a reference method, an implementation of the MUSIC algorithm was provided, as well as ground-truth location of the speakers. We evaluate our algorithm on the azimuth estimation only. The results of the LOCATA test are shown for the single source tracking task in Fig. 3.4 and for the two source tracking task in Fig. 3.5. The proposed method clearly outperforms MUSIC in both tasks, as can be deduced from the inspection of the probability maps and from the score values. The differences are more pronounced in the two speakers case, for which the MUSIC algorithm performs poorly.



Independent.

Parallel HMM.

Coupled HMM.

Figure 3.1: Graphical representation of the Instantaneous model (top) Parallel HMM (middle) and the Coupled HMM (bottom).

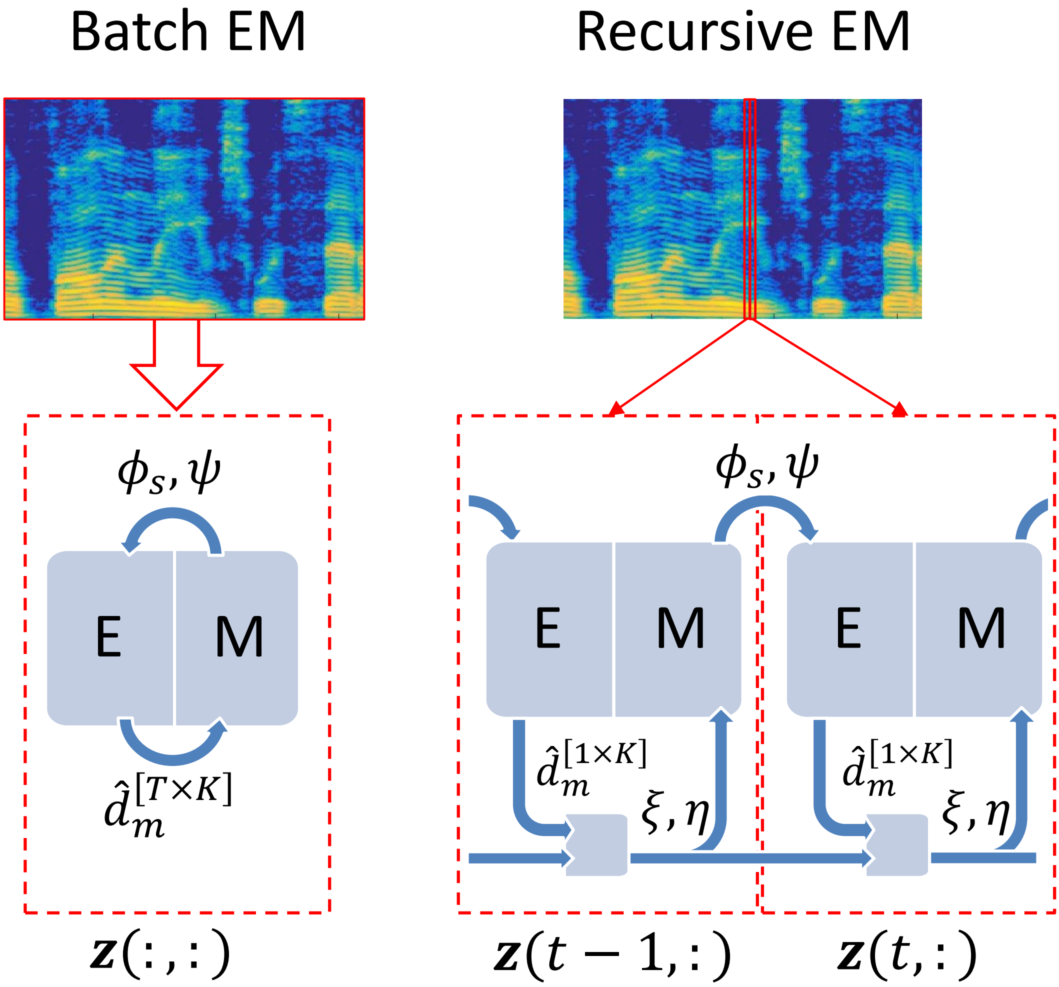
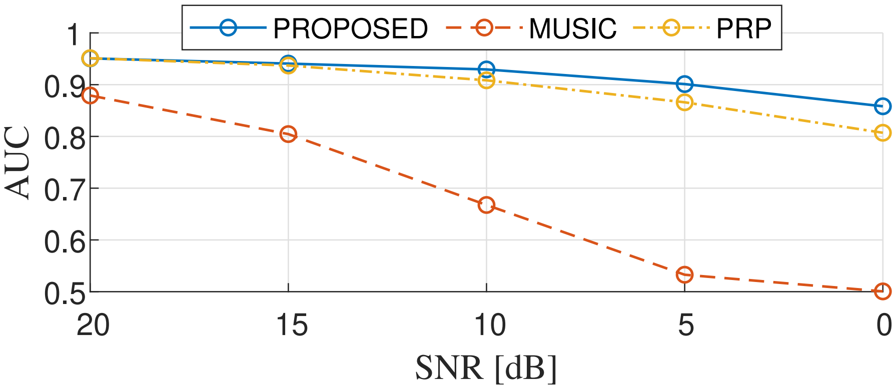
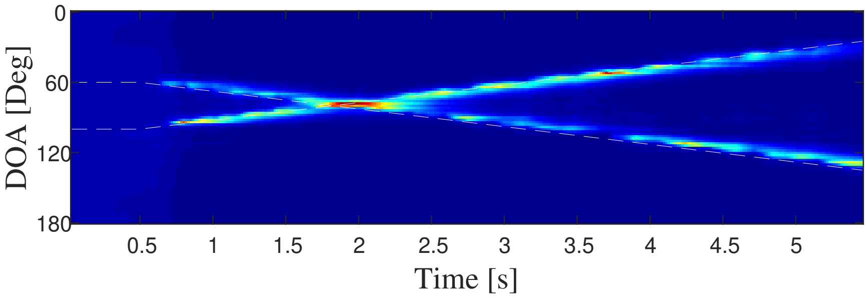


Figure 3.2: Batch and recursive EM illustration.

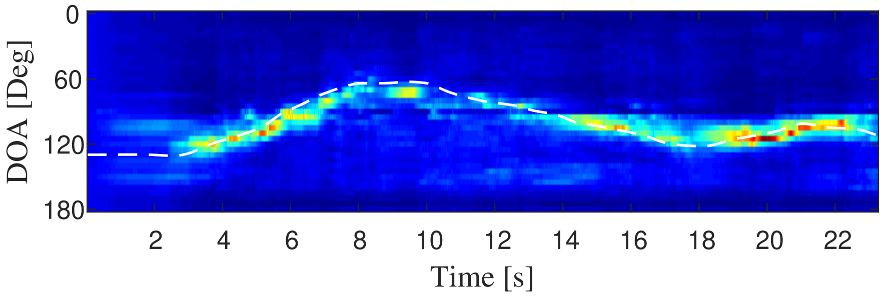


(a) AUC vs. SNR for the instantaneous model and for the reference methods.

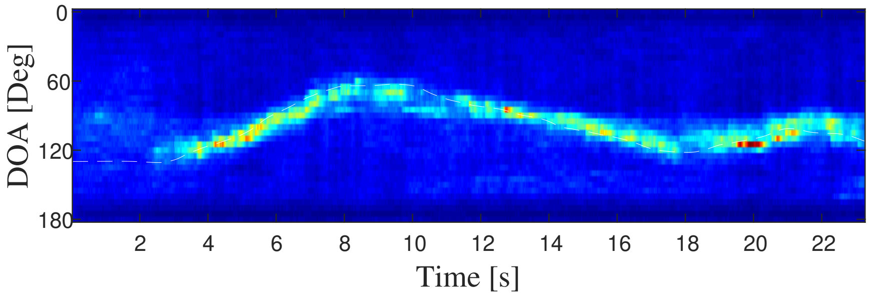


(b) An example probability map for SNR $= 25$ dB and sources velocities $\pm 15 \frac{\text{deg}}{\text{s}}$. The dashed line is the ground truth DOA. The obtained AUC ≈ 0.96.

Figure 3.3: Experimental results of the instantaneous model for simulated data.

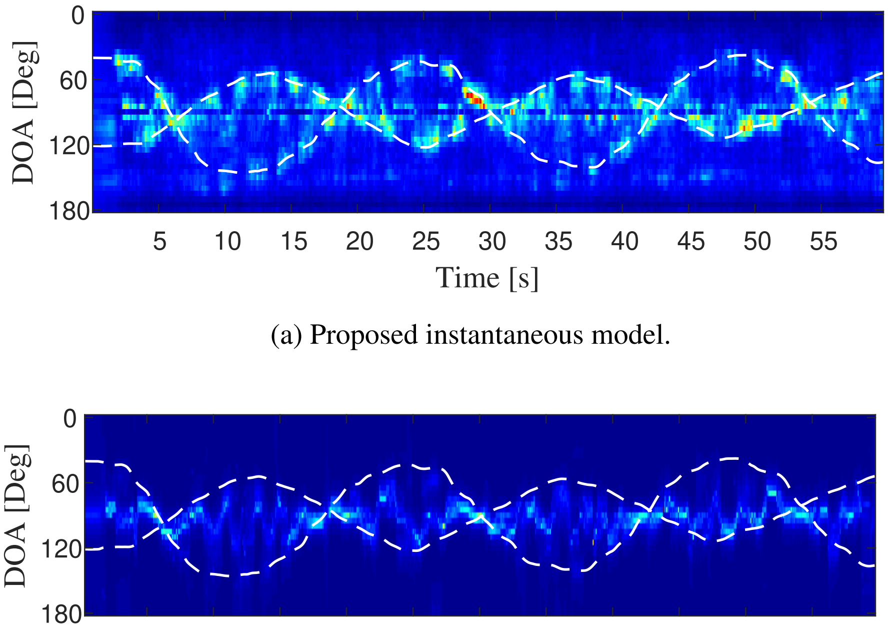


(a) Proposed instantaneous model.



(b) PRP-REM algorithm [10]

Figure 3.4: Probability maps for the LOCATA challenge (Task #3 - single moving speaker). The dashed line is the ground truth azimuth, as provided with the LOCATA database. AUC ≈ 0.95 for both methods.



(a) Proposed instantaneous model.

(b) MUSIC.

Figure 3.5: Probability maps for the LOCATA challenge (Task #4 - two moving speakers). The dashed line is the ground truth azimuth, as provided with the LOCATA database. AUC$= 0.82, 0.69$ for the Proposed instantaneous model and for the MUSIC algorithm, respectively.

Chapter 4

Parallel and Coupled HMM

The material presented in this chapter is based on [23]:

> K. Weisberg and S. Gannot, "Multiple speaker tracking using coupled HMM in the STFT domain," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Le Gosier in Guadeloupe, French West Indies, Dec. 2019.

In this section we will present the parallel and coupled HMM models. In these models the hidden data is defined similarly to the instantaneous model (Chapter 3), while introducing dependency between the hidden variables.

4.1 Hidden Markov Model

In this model we formulate the observations per frequency as a HMM process. The HMM is defined by three probabilities. The first is the emission probability $P(\mathbf{z}_{t,k}|b_{t,k})$ which, following (2.18), is proportional to $T_{t,k}(b_{t,k})$. The second is the transition probability $P(b_{t,k}|b_{t-1,k})$, commonly described by a $M \times M$ transition matrix with elements $\mathbf{A}_{m_1,m_2} = P(b_{t,k} = m_2|b_{t-1,k} = m_1)$, which is the probability of the (t,k)th observation to be associated with candidate DOA m_2, given that the $(t-1,k)$th observation was associated with the DOA candidate m_1. In the single speaker case, the values $\mathbf{A}_{m_1,m_2}$

are set to allow only small changes (or no change) in the DOA readings. In the multiple speaker case, larger changes are also allowed to enable speaker switching. The third probability is the initial-state probability $P(b_{1,k})$, which is set as a uniform distribution for all frequencies.

Under this model, the joint probability of the observations and the hidden data for the kth frequency bin is given by [38]:

$$\begin{aligned} P(\mathbf{z}_{1,k}, \mathbf{z}_{2,k}, \ldots, \mathbf{z}_{T,k}, b_{1,k}, b_{2,k}, \ldots, b_{T,k}) = \\ P(b_{1,k}) \left[\prod_{t=2}^{T} P(b_{t,k}|b_{t-1,k}) \right] \left[\prod_{t=1}^{T} P(\mathbf{z}_{t,k}|b_{t,k}) \right]. \end{aligned} \tag{4.1}$$

Standard HMM inference addresses two questions: 1) what is the most probable state sequence given the observations? 2) what is the marginal posterior of the hidden process given the entire set of observations? While for answering the first, the Viterbi algorithm is applied, providing *hard* estimation of the hidden variables, for answering the second, the FB algorithm is applied, providing a *soft* estimation. In this work, we prefer the latter, which is better suited for aggregating the estimates from all frequencies into a frame-wise DOA estimate.

The FB inference algorithm [38] is based on two terms which are calculated inductively: 1) $\alpha(b_{t,k})$, the *forward* term, and 2) $\beta(b_{t,k})$, the *backward* term. The marginal posterior of the hidden data is then given by:

$$P(b_{t,k}|\mathbf{z}) \propto \alpha(b_{t,k})\beta(b_{t,k}), \tag{4.2}$$

where $\mathbf{z} = \mathrm{vec}_{t,k}\{\mathbf{z}_{t,k}\}$. Note that since the forward and backward terms are not normalized, we can use $T_{t,k}(b_{t,k})$ as the emission probability, rather than the full conditional probability, see (2.18). The forward term is therefore given by:

$$\alpha(b_{1,k}) = P(b_{1,k})T_{1,k}(b_{1,k}) \tag{4.3a}$$

$$\alpha(b_{t,k}) = T_{t,k}(b_{t,k}) \sum_{b_{t-1,k}} \alpha(b_{t-1,k}) P(b_{t,k}|b_{t-1,k}) \tag{4.3b}$$

and the backward term by:

$$\beta(b_{T,k}) = 1 \ \forall b_{T,k} \tag{4.4a}$$

$$\beta(b_{t-1,k}) = \sum_{b_{t,k}} \beta(b_{t,k}) T_{t,k}(b_{t,k}) P(b_{t,k}|b_{t-1,k}). \tag{4.4b}$$

This inference is applied independently to each frequency. The model, denoted *parallel* HMM, is depicted in Fig. 3.1 (middle).

4.2 Coupled Hidden Markov Model

The coupled HMM is an extension of the parallel HMM, with the state of each process depending also on the states of the other processes from the previous time-step. In order to simplify the inference, specific structure is commonly assumed in which the conditional distribution is a linear combination of the marginal dependencies [25]:

$$P(b_{t,k}|b_{t-1,1}, \dots, b_{t-1,K}) = \sum_{k'=1}^{K} \mathbf{B}_{k,k'} P_{k,k'}(b_{t,k}|b_{t-1,k'}), \tag{4.5}$$

where $\mathbf{B}$ is a coupling matrix between frequency pairs, and $P_{k,k'}(b_{t,k}|b_{t-1,k'})$ is the transition probability between the states at time-step $t-1$ and time-step t, which in general, also depends on the frequencies k' and k at time-steps $t-1$ and t, respectively. However, in this work we simplify the transition to a frequency-independent matrix, namely $P_{k,k'}(b_{t,k} = m_2|b_{t-1,k'} = m_1) = P(b_{t,k} = m_2|b_{t-1,k'} = m_1) = \mathbf{A}_{m_1,m_2}$. The coupled HMM is depicted in Fig. 3.1 (bottom).

In order to find the posterior of each of the hidden variables given the entire set of observations $P(b_{t,k}|\mathbf{z})$, one should use the FB algorithm. An exact inference of this model may be obtained by constructing a M^K-dimension compound state comprising all frequencies. The forward and backward variables will also comprise all frequencies, namely $\alpha(b_{t,1},\ldots,b_{t,K})$ and $\beta(b_{t,1},\ldots,b_{t,K})$. To simplify the inference procedure, we will use decomposed variables: $\alpha(b_{t,1}),\ldots,\alpha(b_{t,K})$ and $\beta(b_{t,1}),\ldots,\beta(b_{t,K})$. This decomposition may be implemented in several ways [39; 40]. In the current contribution, we preferred to use the simple approximation, which was shown to yield a satisfactory posterior [26]. This was also verified in our experimental study. In this method, the initialization of the forward and backward variables is similar to (4.3a) and (4.4a) and the recursive inference of the forward variable is given by:

$$\alpha(b_{t,k}) = T_{t,k}(b_{t,k}) \sum_{k'=1}^{K} \mathbf{B}_{k,k'} \sum_{b_{t-1,k'}} \alpha(b_{t-1,k'}) P(b_{t,k}|b_{t-1,k'}) \tag{4.6}$$

and, similarly, for the backward variable:

$$\beta(b_{t-1,k}) = \sum_{k'=1}^{K} \mathbf{B}_{k,k'} \sum_{b_{t,k'}} \beta(b_{t,k'}) T_{t,k'}(b_{t,k'}) P(b_{t,k}|b_{t-1,k'}). \tag{4.7}$$

Note that if the entries of the coupling matrix are set to $\mathbf{B}_{k,k'} = \delta(k',k)$, with $\delta(\cdot,\cdot)$ the Kronecker delta function, the coupled HMM collapses to the parallel HMM. On the contrary, we observed that coupling all frequencies together (e.g. by setting $\mathbf{B}_{k,k'} = \frac{1}{K}$ $\forall k,k'$) tends also to couple all DOA estimates across frequencies. While this is a desirable property in the single-speaker case, it falls short in modelling the multiple-speaker case, where different sets of frequencies may be associated with different speakers. We therefore propose to *couple* only the processes related to neighboring frequencies, namely the DOA of the (t,k)th bin depends on the $(t-1,k)$th and $(t-1,k\pm 1)$th bins, and to set all other coupling coefficients to zero. Coupling more frequencies did not result in further improvement.

As a result of the application of the FB algorithm, the posterior of the hidden data is obtained. As this is a frequency-wise soft decision, it should be aggregated along the frequency-index to obtain a single decision per frame. An intuitive approach is to average the soft associations of the frequencies to each time frame, namely:

$$\psi_t(m) = \frac{\sum_k \widehat{b}_{t,k}(m)}{K} \tag{4.8}$$

where $\widehat{b}_{t,k}(m) = P(b_{t,k} = m|\mathbf{z})$. Then, any peak-picking method can be applied to find the actual DOA of all speakers.

4.3 Experimental study

The proposed algorithm was evaluated using two datasets: simulated time-varying scenes, generated by a signal generator, and real multichannel audio recordings from the LOCATA challenge [36].

4.3.1 Algorithm settings and baseline methods

The signals were resampled to 16 kHz and transformed into the STFT domain with frame-length of 64 ms and 75% overlap. The frequency band used for localization was $300 - 4500$ Hz. For applying the algorithm, a grid of possible azimuth angles is required. We used a grid between $-90°$ and $90°$, with a resolution of $2°$.

The entries of the transition matrix $\mathbf{A}$ were set to:

$$\log \mathbf{A}_{m_1,m_2} \propto \begin{cases} 20 \text{ if } m_1 \in [m_2-1, m_2, m_2+1] \\ 0 \text{ otherwise} \end{cases} \tag{4.9}$$

and the coupling matrix $\mathbf{B}$, were set to:

$$\mathbf{B}_{k_1,k_2} \propto \begin{cases} 10^3 \text{ if } k_1 = k_2 \\ 1 \text{ if } k_1 \in [k_2-1, k_2+1] \\ 0 \text{ otherwise} \end{cases} \tag{4.10}$$

	35 dB	25 dB	15 dB
MUSIC [2]	0.8728	0.8535	0.8114
Instantaneous model	0.9374	0.9303	0.9081
Parallel HMM	0.9440	0.9327	0.9265
Coupled HMM	**0.9443**	**0.9361**	**0.9288**

Table 4.1: The AUC score results for the simulation study of two moving speakers. Best value for AUC is 1 and the lowest value 0.5.

where the sign $\propto$ stands for proportion.

Similar to the instantaneous model, the proposed method provides a probability map as a function of the time-step rather than a direct estimate of the DOAs. For estimating the actual trajectory of the speakers, one can use any peak-picking method. To circumvent the effects of the specific method selected, we will report instead the AUC figures. For establishing the ROC curve, DOA estimates in the range of $\pm 3^\circ$ around the true value are considered *true positive* and otherwise considered *false positive*.

For baseline methods, we used both the MUSIC algorithm [2], as provided by the LOCATA challenge, as well as instantaneous model (chapter 3) with smoothing parameters $\gamma_\psi = 0.1$ and $\gamma_{\phi_s} = 0.8$. For a fair comparison, the proposed results and the MUSIC results were smoothed and normalized to obtain a pseudo-distribution similar to [22].

4.3.2 Evaluation using Simulations

In the simulated scenario, clean anechoic speech signals were drawn from the TIMIT database [37], where speech utterances of the same speaker were concatenated to obtain a 4 s long speech signal. The speakers were randomly selected from 26 different speakers. The simulated signals were generated using a signal generator [41]. The room dimensions were set to $6 \times 4 \times 3$ m with reverberation time $T_{60} = 650$ ms. The signals were captured by an eight-microphone linear array with inter-distances of $[3, 3, 3, 8, 3, 3, 3]$ cm from one another, together with an additive spatially-diffuse speech-like noise with various SNR values. The noise covariance matrix $\Phi_{\mathbf{v},k}$ was estimated using the long speech-absent segments.

	Task #3	Task #4
MUSIC [2]	0.8844	0.7756
Instantaneous model	0.9455	0.8608
Parallel HMM	**0.9589**	**0.8622**
Coupled HMM	0.9547	0.8601

Table 4.2: The AUC results for the LOCATA experiment. Task #3 is single moving speaker and task #4 is two moving speaker.

Twenty Monte-Carlo trials, simulating two moving sources scenarios, were examined. In each scenario, the initial DOAs of the speakers were set to 40° and 100°, respectively. The sources moved from their initial positions in a circle with a radius of 1 m around the array center and with angular velocity $[15, -15]\ \frac{\text{deg}}{\text{s}}$, respectively. An example probability map from this experiment is shown in Fig. 4.1 (top). The results of this experiment are depicted in Table 4.1.

4.3.3 Evaluation on LOCATA dataset

The data for the LOCATA challenge [36] were recorded in a room of size $7.1 \times 9.8 \times 3$ m with a reverberation time $T_{60} \sim 0.55s$. We tested our algorithm on Task #3, which is a recording of a single moving speaker, and Task #4, which is a recording of two moving speakers. We used the data recorded by the linear array (DICIT). We used the first recording (Recording #1) of each task. In order to estimate the noise covariance matrix, we used noise-only frames $\approx$ 2 s long from the beginning of each utterance. For the multiple speakers case, we used $\mathbf{A}$ and $\mathbf{B}$ as in Sec. 4.3.1, while for the single speaker case we kept the same coupling matrix $\mathbf{B}$ but modified the transition matrix $\mathbf{A}$ to $\log \mathbf{A}_{m_1,m_2} \propto 20$ if $m_1 \in [m_2 - 1, m_2, m_2 + 1]$ and $-\infty$ otherwise. For calculating the AUC, we used the ground-truth location of the speakers, as provided by the challenge. We evaluated our algorithm on the azimuth estimation only. The LOCATA experiment results are shown in Fig. 4.1 (middle and bottom). Comparative study for the proposed methods and the baseline methods can be found in Table 4.2.

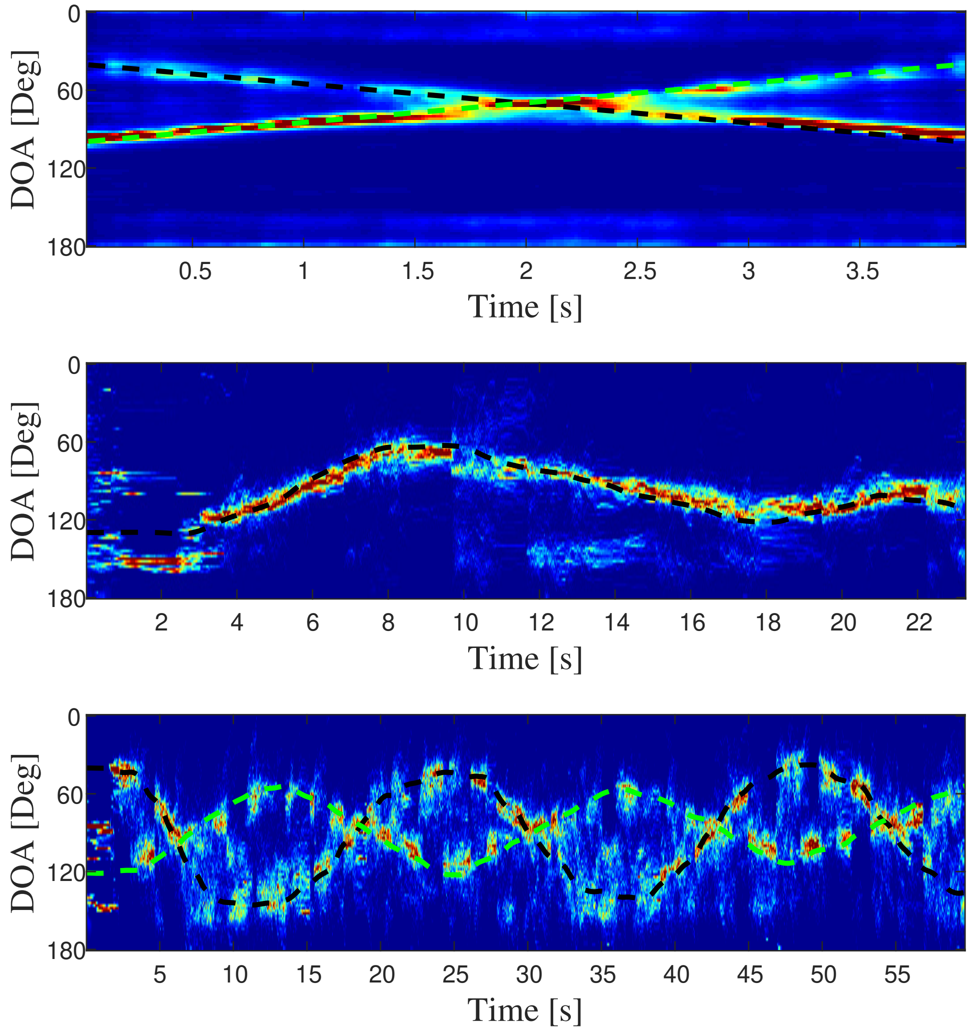


Figure 4.1: Probability maps for the simulation experiment (top) and the LOCATA challenge - Task #3 (Single Speaker, middle), and Task #4 (Two Speakers, bottom).

Chapter 5

Factor Graph Model

The material presented in this chapter is based on [24]:

> K. Weisberg, B. Laufer-Goldshtein, and S. Gannot, “Simultaneous tracking and separation of multiple sources using factor graph model,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2848–2864, 2020.

In this chapter we present the factor graph model. In this model, the hidden data is defined to be both the DOAs of the speakers and the associations of the TF bins to the speakers. For inference we derive novel inference algorithm based on the LBP algorithm.

5.1 The model

In this model, we consider the speaker associations $a_{t,k}$ and the DOAs $d_t(j)$ as latent variables that we would like to infer from the observations $\mathbf{z}_{t,k}$. Applying Bayes rule, the posterior of the latent variables is given by:

$$P(\mathbf{d}, \mathbf{a}|\mathbf{z}) = \frac{P(\mathbf{z}|\mathbf{a}, \mathbf{d})P(\mathbf{a})P(\mathbf{d})}{P(\mathbf{z})} \tag{5.1}$$

where $\mathbf{a} = \mathrm{vec}_{t,k}\{a_{t,k}\}$, $\mathbf{d} = \mathrm{vec}_{t,j}\{d_t(j)\}$, $\mathbf{z} = \mathrm{vec}_{t,k}\{\mathbf{z}_{t,k}\}$, and we assume independence between the DOAs $\mathbf{d}$ and the associations $\mathbf{a}$.

The main task is to find the marginal posterior of the variables, namely $P(a_{t,k}|\mathbf{z})\ \forall t,k$, and $P(d_t(j)|\mathbf{z})\ \forall t,j$. However, an exact computation of these marginal distributions is intractable. In [18] this posterior was approximated by a product of probabilities from known families, and the variational inference was used for estimating the parameters of these probabilities. In the current work, we present a statistical model in which the posterior is given in a form of a factor graph. We then propose to use the LBP inference algorithm in order to find the marginal posterior for each variable.

In this section, we define the prior probabilities of the hidden variables $P(\mathbf{a})$ and $P(\mathbf{d})$, as well as the probability of the observations given the hidden variables $P(\mathbf{z}|\mathbf{a},\mathbf{d})$, and use them to form the factor graph of the posterior probability (5.1). The inference algorithm that is applied to this factor graph model is described in Section 5.2. A brief general review on factor graph models and their inference methods is given in Appendix A.

5.1.1 The DOA model

Following [17; 18] the prior probabilities of the DOAs of each of the speakers are modeled as separated and independent Markov chains. The state of the Markov process associated with each speaker is the DOA index of the corresponding speaker at each-time step. The transition probabilities are set in a way that allows the DOA of each speaker to vary smoothly overtime. Accordingly, the joint probability of $\mathbf{d}$ is given by:

$$P(\mathbf{d}) = \prod_{j=1}^{J}\left[\Omega_j(d_1(j))\prod_{t=2}^{T}\Psi(d_{t-1}(j),d_t(j))\right] \tag{5.2}$$

where we have defined the following *potential* functions:

$$\Psi(m_1,m_2) = P(d_t(j)=m_2|d_{t-1}(j)=m_1) \tag{5.3a}$$

$$\Omega_j(m) = P(d_1(j)=m) \tag{5.3b}$$

where $P(d_t(j) = m_2|d_{t-1}(j) = m_1)$ is the probability to switch from one DOA to another in subsequent time steps, and $P(d_1(j) = m)$ is the initial probability of the jth speaker at time $t = 1$. In order to achieve a continuous trajectory, the transition probability is set as:

$$P(d_t(j)|d_{t-1}(j)) \propto \begin{cases} 1 \text{ if } d_t(j) = d_{t-1}(j) \\ \exp(-\alpha) \text{ if } d_t(j) = d_{t-1}(j) \pm 1 \\ 0 \text{ otherwise} \end{cases} \tag{5.4}$$

where $\alpha > 0$ is a hyper-parameter which controls the smoothness of the trajectory. The initial DOA probability is assumed to be known. However, we observed in our experiments (see Sec. 5.3.3) for a case with three speakers) that it may also be randomly initialized, hence a prior knowledge on the initial DOA is in practice unnecessary.

5.1.2 The association model

For the prior probability of the association variables $\mathbf{a}$, we propose two alternative models. The simple model is an i.i.d. distribution where an independence between the associations in different TF bins is assumed, and each of them is uniformly distributed, namely:

$$P(\mathbf{a}) = \prod_{t,k} \frac{1}{J} = \frac{1}{J^{TK}} \tag{5.5}$$

which is a constant expression. In the following, we derive the inference algorithm for this model.

An alternative model is described in Section 5.2.6 following [9]. This model takes into account the speech activity pattern across time and frequency, and represents the relation between adjacent TF bins using a Markov random field (MRF). The MRF model provides a more accurate description of the behavior of the association variables across time and frequency compared to the uniform model (5.5), at the cost of slightly increasing the complexity of the inference scheme. In the experimental part in Section 5.3, we show that the MRF model has a slight advantage over the uniform model in terms of the actual performance. By describing both models, we would like to further demonstrate the flexibility of the proposed statistical framework that facilitates the use of various models for the associations with only small adjustments to the proposed inference algorithm.

5.1.3 The observation factor

For the factor graph model, we need to explicitly define a factor for each observation as a function of all the associated latent variables. Thus, we rewrite (2.19) as:

$$P(\mathbf{z}|\mathbf{a},\mathbf{d}) = \frac{1}{C_z}\prod_{t,k}\Upsilon_{t,k}(a_{t,k}, d_t(1)\dots d_t(J)) \tag{5.6}$$

where $\frac{1}{C_z} \equiv \prod_{t,k} G_{t,k}$ is a constant normalization and:

$$\Upsilon_{t,k}(a_{t,k}, d_t(1)\dots d_t(J)) \equiv T_{t,k}(d_t(a_{t,k})). \tag{5.7}$$

We denote this function as the *observation* factor. Note that while $T_{t,k}(\cdot)$ is a function of a single variable $d_t(a_{t,k}) \in [1\dots M]$, the potential function $\Upsilon_{t,k}(\cdot,\dots,\cdot)$ is a function of $J+1$ variables, namely, $a_{t,k}$ and $d_t(1)\dots d_t(J)$. The definition of $\Upsilon_{t,k}(\cdot,\dots,\cdot)$ is necessary as the factor graph model requires that the factors are presented as direct functions of each of the individual hidden variables separately. Note also that in contrast to the DOA factor Ψ (5.4), which is fixed along time, the observation factor varies across time and frequency, since it is determined by the specific observation in each TF bin.

5.1.4 The Factor Graph

We can now express the posterior $P(\mathbf{a},\mathbf{d}|\mathbf{z})$ as a factor graph. Substituting (5.2), (5.5) and (5.6) into (5.1), we obtain:

$$P(\mathbf{d},\mathbf{a}|\mathbf{z}) = \frac{1}{C}\prod_{t,k}\Upsilon_{t,k}(a_{t,k}, d_t(1)\dots d_t(J))\prod_{j=1}^{J}\Omega_j(d_1(j))\prod_{t=2}^{T}\Psi(d_{t-1}(j), d_t(j)) \tag{5.8}$$

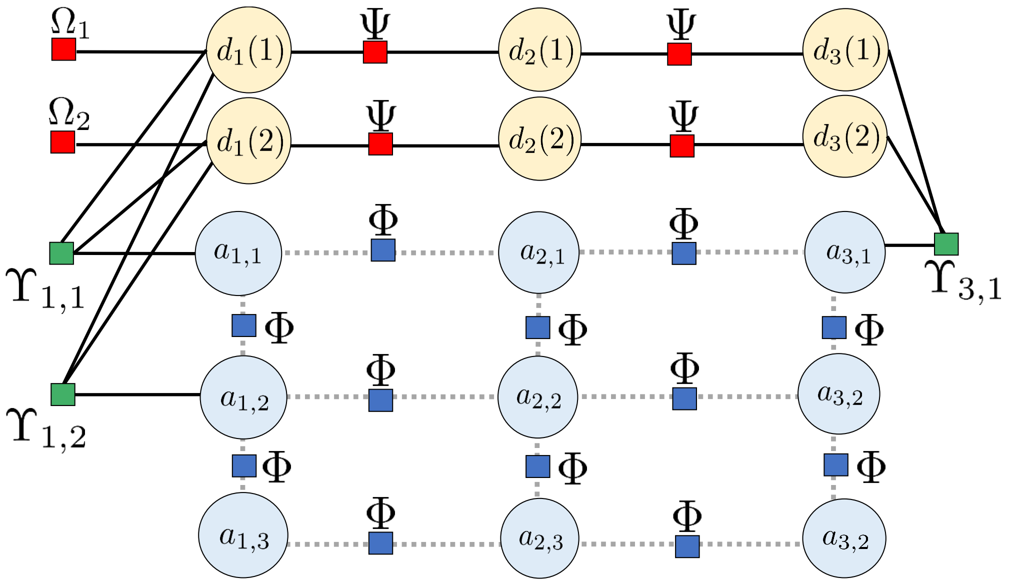


Figure 5.1: The proposed factor graph. Here, $J = 2$ speakers, $K = 3$ frequencies, and $T = 3$ time-frames, for simplicity. Only three out of $T \times K$ observation factors are drawn. The gray dashed lines and the factors Φ correspond to the modified factor graph presented in Section 5.2.6, which is based on the MRF model for the associations. For the uniform distribution model of the associations (5.5) these connections and factors are ignored.

where the factors $\Psi(\cdot, \cdot)$, $\Omega_j(\cdot)$ and $\Upsilon_{t,k}(\cdot, \ldots, \cdot)$ are defined in (5.3a),(5.3b) and (5.7) respectively, and $C \equiv C_z \cdot J^{TK} \cdot P(\mathbf{z})$ is a normalization constant. The factor graph model is illustrated in Fig. 5.1.

5.2 Inference using the LBP

The obtained factor graph contains loops, as can be seen in the illustrative example in Fig. 5.1, and therefore the loopy belief propagation (LBP) [42] can be used for its inference. In this section, we derive the LBP algorithm to approximate the marginal posteriors of the latent variables given the observations. The final DOA trajectory and the separated signals are then obtained based on the computed marginals. In the LBP, messages are sent from the factors to the variables and vice versa (see Appendix A). In the proposed model there are three groups of factors: i) Ω (connected to $d_1(1), \ldots, d_1(J)$); ii) Ψ (connected to $\mathbf{d}$); and iii) Υ (connected to all variables). The messages are functions of the corresponding variable (either source or destination), and are calculated using the general

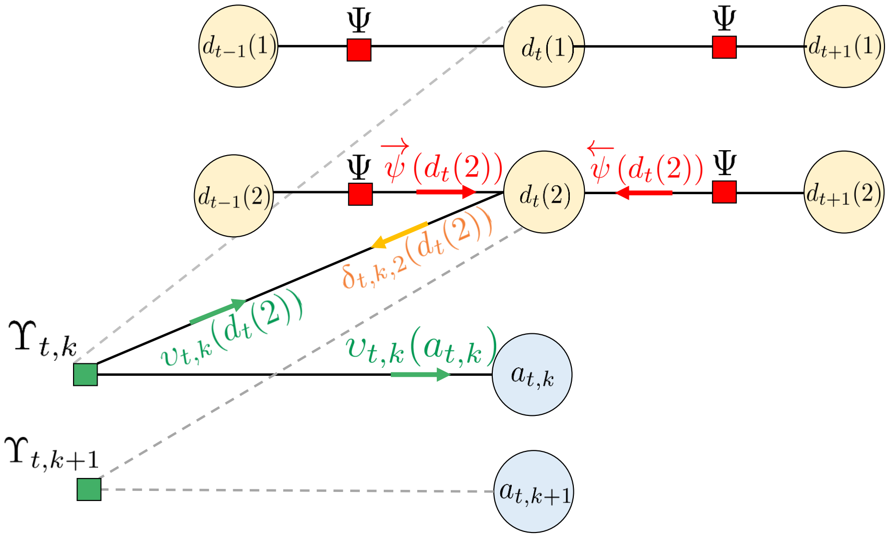


Figure 5.2: The messages in the proposed LBP algorithm. The arrows are pointing from the sending variable/factor to the receiving variable/factor, and the notation of the associated message is written above/below the arrow.

equations (A.2a) and (A.2b). However, these general equations can be simplified in our case to achieve more efficient formulas, as shown in the sequel.

5.2.1 Notation

In the following derivations we use a simplified set of notations. The messages from Ψ to $d_t(j)$ are denoted by $\overrightarrow{\psi}(d_t(j))$ and $\overleftarrow{\psi}(d_t(j))$ for the forward and backward messages, respectively. For the completeness of the notation we use this notation also for $t = 1$ and $t = T$, where for $t = 1$ the forward message of the factor Ψ is replaced with the corresponding Ω factor, and for $t = T$ the backward message is fixed to uniform, as there is no backward message to the last variable. For the observation factor, we use $\upsilon_{t,k}(\cdot)$ for the outgoing messages from the observation factor to each of the variables connected to it, where the destination variable is deduced from the term in the brackets, i.e. $\upsilon_{t,k}(d_t(j))$ refers to messages to the DOA variables and $\upsilon_{t,k}(a_{t,k})$ refers to messages to the association variables. The messages from $d_t(j)$ to the observations are denoted by $\delta_{t,k,j}(d_t(j))$. The different types of messages are illustrated in Fig. 5.2.

5.2.2 Messages from the DOA factors

In general the factors send messages to their neighbor variables (the *outgoing messages*), where these messages depend on the incoming messages from variables to the factors (the *incoming messages*). However, for Ψ and Ω, the factor is a function of only a single or two variables and it has only a single incoming message. Therefore, we do not explicitly define the incoming messages for these factors. Instead, we substitute the incoming message with its definition (A.2a). As a result, each of the outgoing messages is expressed in terms of the outgoing messages of its neighbor factors to the corresponding variable.

The forward messages of Ψ for $t > 1$ are given by:

$$\overrightarrow{\psi}(d_t(j)) = \sum_{d_{t-1}(j)} \Psi(d_{t-1}(j), d_t(j)) \overrightarrow{\psi}(d_{t-1}(j)) \overline{\upsilon}_{t-1}(d_{t-1}(j)) \tag{5.9}$$

where

$$\overline{\upsilon}_t(d_t(j)) = \prod_k \upsilon_{t,k}(d_t(j)) \tag{5.10}$$

is the message of all K observations to $d_t(j)$. For $t = 1$ the message is given by:

$$\overrightarrow{\psi}(d_1(j)) = \Omega_j(d_1(j)). \tag{5.11}$$

The backward message $\overleftarrow{\psi}(d_t(j))$ is symmetric, where for $t = T$ it is set to uniform for completeness.

5.2.3 Message from and to the observation factors

The incoming messages from the DOA variables $d_t(j)$ to the observations Υ are given by the multiplication of the incoming messages of each DOA variable (A.2a), namely:

$$\delta_{t,k,j}(d_t(j)) = \overrightarrow{\psi}(d_t(j)) \overleftarrow{\psi}(d_t(j)) \prod_{\tilde{k} \neq k} \upsilon_{t,k}(d_t(j)). \tag{5.12}$$

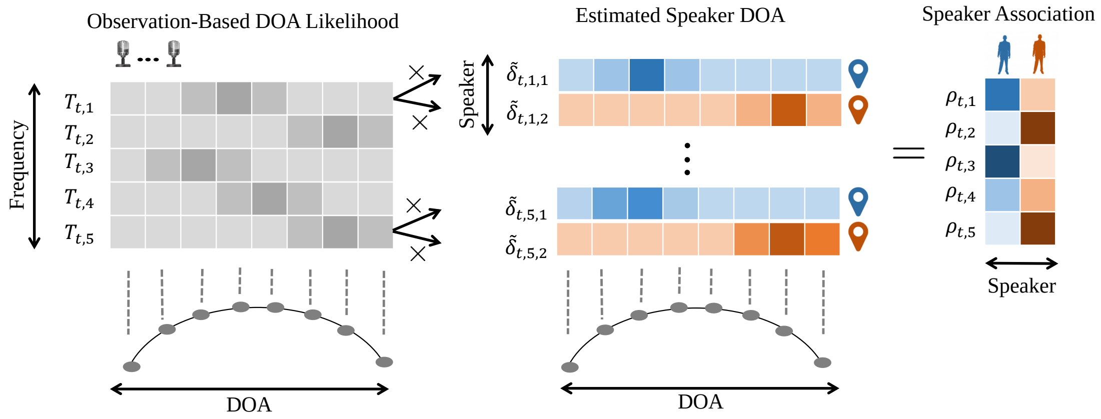


Figure 5.3: Illustration of the calculation of $\rho_{t,k}(j)$. The vectors represent the probabilities over the candidate DOAs, where darker elements correspond to more probable candidates. The case of two speakers is illustrated by blue and orange vectors representing the current DOA estimate of the each of the speakers. The observation-based DOA likelihood vectors (in gray) are correlated with the estimated DOA of the speakers, resulting in $\rho_{t,k}(j)$, which represents the association of the (t, k)th bin to either of the speakers based on the observation.

The full derivation of the outgoing messages from the observation factors to their neighbor variables can be found in Appendix B. In order to simplify the messages, we first define the correlation between $T_{t,k}(:)$ and the normalized incoming message $\delta_{t,k,j}(:)$ as:

$$\rho_{t,k}(j) = \sum_{m=1}^{M} T_{t,k}(m)\tilde{\delta}_{t,k,j}(m) \tag{5.13}$$

where $\tilde{\delta}_{t,k,j}(m) = \frac{\delta_{t,k,j}(m)}{\sum_m \delta_{t,k,j}(m)}$ is the normalized message. The correlation measures the similarity between $\delta_{t,k,j}(:)$, which is the current estimate of the jth speaker DOA, and $T_{t,k}(:)$, which is the (t, k)th bin DOA likelihood based on the observation. The obtained $\rho_{t,k}(j)$ is therefore a non-normalized association of the (t, k)th bin to a speaker based on the similarity between the observed DOA and the estimated DOA of each of the speakers, namely, a higher value is given to the speaker whose estimated DOA matches the observed DOA, and vice versa. This process is illustrated in Fig. 5.3.

Using the definition of $\rho_{t,k}(j)$, the message from the observation factor to the association variable is given by:

$$\upsilon_{t,k}(a_{t,k}) = \rho_{t,k}(a_{t,k}) \tag{5.14}$$

and the message from the observation factor to the DOA variables is given by:

$$\upsilon_{t,k}(d_t(j)) = T_{t,k}(d_t(j)) + \sum_{\ell \neq j} \rho_{t,k}(\ell). \tag{5.15}$$

The meaning of the message conveyed by Υ to the jth speaker DOA is as follows. The message consists of two terms: $T_{t,k}(d_t(j))$ that depends on the DOA value $d_t(j)$, and $\sum_{\ell \neq j} \rho_{t,k}(\ell)$, which is independent of $d_t(j)$. If one of the other speakers is active with high probability at this TF bin, then the value of the second term is high, and the message is close to uniform with respect to $d_t(j)$, i.e. does not indicate any preference to a certain DOA. Otherwise, the jth speaker is probably active at this TF bin, and the message is dominated by the first term $T_{t,k}(d_t(j))$, which is the DOA likelihood based on the (t,k)th bin observation.

In the next step, the messages from all frequencies are integrated together for each speaker in $\overline{\upsilon}_t(d_t(j))$ (5.10) to determine its new DOA. In this integration, uniform messages do not add any information. Therefore the integrated message for the jth speaker, contains only the information from the relevant frequencies where the jth speaker is active. The calculation of the messages $\overline{\upsilon}_t(d_t(j))$ is illustrated in Fig. 5.4.

Note that while the message to the variable $a_{t,k}$ depends on the incoming messages from all other variables $\rho_{t,k}(1), \ldots, \rho_{t,k}(J)$, the message to the DOA variable $d_t(j)$ of the jth speaker depends on the message from all other variables $\rho_{t,k}(1), \ldots, \rho_{t,k}(j-1), \rho_{t,k}(j+1), \ldots, \rho_{t,k}(J)$ except for the jth speaker message $\rho_{t,k}(j)$, since by the definition of the LBP algorithm, the message to a particular variable depends on all incoming messages except for the message from this variable itself.

Three additional notes on the differences between the general formulation of the message (A.2b) in Appendix A and the simplified message (5.15) are in place. 1) Instead of the raw incoming messages $\delta_{t,k,1}(:), \ldots, \delta_{t,k,J}(:)$, the outgoing messages use $\rho_{t,k}(1), \ldots, \rho_{t,k}(J)$ defined by the correlation between the incoming messages and the observations (5.13); 2) The message from the association variable a does not appear here

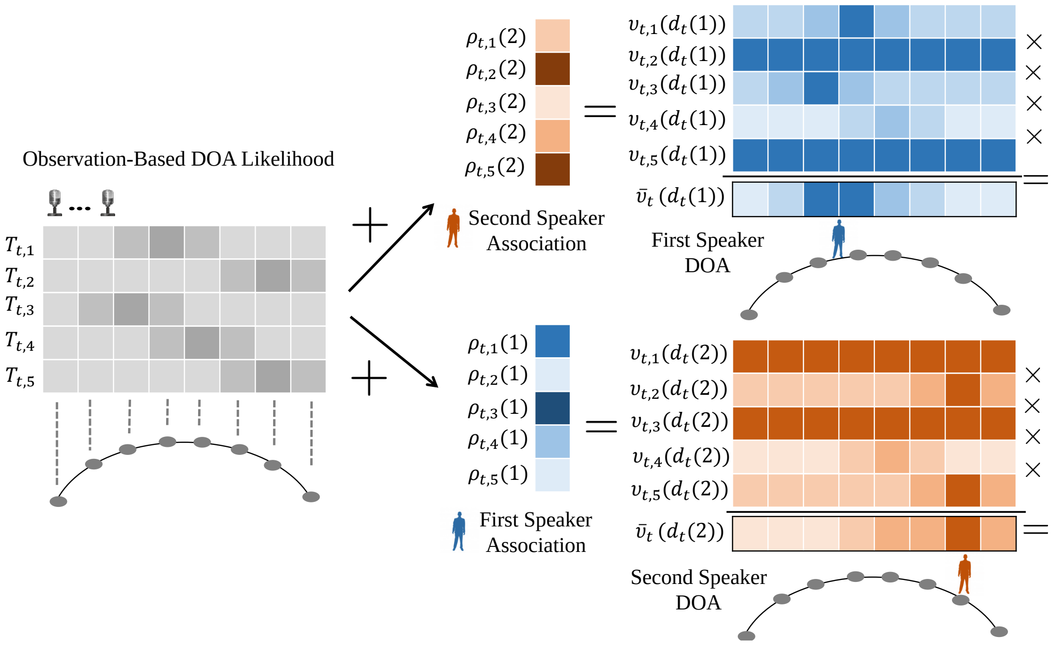


Figure 5.4: Illustration of the calculation of $\overline{\upsilon}_t(d_t(j))$. Darker elements correspond to higher values. For each speaker the association to the other speakers (colored in orange or blue) is added to the DOA likelihood (colored in gray). The result is the per-frequency non-normalized probability for each speaker $\upsilon_{t,k}(d_t(j))$. Multiplication along the frequencies, results in the non-normalized DOA distribution.

since this variable has no connected factor except the observation; and 3) The obtained messages involve only $T_{t,k}(:)$, and not the entire factor $\Upsilon_{t,k}$, since this is all the information that the factor contains (5.7).

5.2.4 The inference algorithm

The full inference algorithm is as follows. We first initialize all messages to be uniform, then we iterate over all the variables and update their incoming messages from their associated factors using equations (5.9, 5.11, 5.12, 5.14, 5.15). The iterations of the LBP algorithm are stopped when the following stopping criterion is satisfied: the maximum change in the log messages between subsequent iterations is smaller than ε or when the number of iterations reaches $N_{\max}$, which is defined as the maximum number of iterations.

The final stage is to compute the marginals, using the following equations:

$$P(d_t(j)|\mathbf{z}) \propto \overrightarrow{\psi}(d_t(j))\overleftarrow{\psi}(d_t(j))\overline{\upsilon}_t(d_t(j)) \tag{5.16a}$$

$$P(a_{t,k}|\mathbf{z}) \propto \upsilon_{t,k}(a_{t,k}) \tag{5.16b}$$

where $\overline{\upsilon}_t(d_t(j))$ is defined in (5.10) and the sign $\propto$ implies that an additional normalization step is required. The inference algorithm is summarized in Algorithm 2.

Algorithm 2 Loopy belief propagation (LBP) for simultaneous tracking and separation

```
Initialize all messages to uniform
while Stopping criterion not satisfied do
    for t=1:T do
        update $\Psi$ messages $\forall j$ using (5.9 or 5.11)
        compute $\tilde{\delta}_{t,k,j}(d_t(j))$ $\forall j, k$ using (5.12)

        compute $\upsilon_{t,k}(a_{t,k})$ and $\upsilon_{t,k}(d_t(j))$ $\forall j, k$ using (5.14,5.15)

    end
end
compute the marginals using (5.16a,5.16b)
```

5.2.5 Tracking and separation

Applying the inference procedure, the marginals of all the hidden variables are computed. The trajectory of each speaker is obtained by selecting the most probable value for each $d_t(j)$:

$$\hat{d}_t(j) = \underset{m \in \{1,\ldots,M\}}{\text{argmax}} \; P(d_t(j) = m|\mathbf{z}). \tag{5.17}$$

The association variables provide the separation mask, which can be used in order to separate the signal to its different sources. Following [15, Eq. (15)], the individual speech signal can be estimated by spatial multichannel filtering followed by single channel post-filtering (see e.g. [43]):

$$\widehat{S}_{t,k}(j) = P(a_{t,k} = j|\mathbf{z}) \hat{s}_{\mathbf{w},t,k}(\hat{d}_t(j)) \tag{5.18}$$

where $P(a_{t,k} = j|\mathbf{z})$ is responsible for enhancing the jth speaker and attenuating the other speakers and $\hat{s}_{\mathbf{w},t,k}(\hat{d}_t(j))$ defined in (2.10) is the output of the MVDR-BF directed towards the estimated DOA of the jth speaker, and is responsible for reducing the ambient noise.

5.2.6 MRF model for the associations

In this section, we replace the uniform model of the association variables (5.5) by a more complex statistical model as suggested in [9], and describe the corresponding modifications to the factor graph and the inference algorithm. It was shown in [9] that in order to smooth the associations, and to reduce musical noise, it is more reasonable to model the dependency between the association variables in adjacent time and frequency indexes using the Markov random field (MRF) model. For this model, the joint probability of the association variables is given by:

$$P(\mathbf{a}) = \frac{1}{C_a} \prod_{t,k} \prod_{\tilde{t},\tilde{k} \in \mathcal{G}\{t,k\}} \Phi(a_{t,k}, a_{\tilde{t},\tilde{k}}) \tag{5.19}$$

where $\mathcal{G}\{t,k\} = \{(t-1,k), (t+1,k), (t,k-1), (t,k+1)\}$ is the group of the indexes couples, C_a is a normalization constant, and $\Phi(j_1, j_2)$ is usually defined as:

$$\Phi(j_1, j_2) = \exp(\beta \delta_K(j_1, j_2)) \tag{5.20}$$

where $\delta_K(\cdot,\cdot)$ is the discrete Kronecker delta function, and $\beta > 0$ is a hyper-parameter of the algorithm. This model encourages nearby TF bins to be associated to the same source, and makes the association map smoother. The parameter β controls this smoothness, where the map becomes smoother as β increases.

Incorporating this model, the factor graph is given by:

$$
\begin{aligned}
P(\mathbf{d},\mathbf{a}|\mathbf{z}) = \frac{1}{C}\prod_{t,k} & \Upsilon_{t,k}(a_{t,k}, d_t(1)\dots d_t(J)) \\
& \prod_{j=1}^{J}\Omega_j(d_1(j))\prod_{t=2}^{T}\Psi(d_{t-1}(j), d_t(j)) \\
& \prod_{t,k}\prod_{\tilde{t},\tilde{k}\in\mathcal{G}\{t,k\}}\Phi(a_{t,k}, a_{\tilde{t},\tilde{k}}).
\end{aligned}
\tag{5.21}
$$

In the LBP we add $\overrightarrow{\phi}_t(a_{t,k})$ and $\overleftarrow{\phi}_t(a_{t,k})$, for the backward and forward messages of the MRF factors Φ in the time dimension and $\overrightarrow{\phi}_f(a_{t,k})$ and $\overleftarrow{\phi}_f(a_{t,k})$, for the messages in the frequency dimension. The outgoing messages of the factor Φ are given by:

$$
\overrightarrow{\phi}_t(a_{t,k}) = \sum_{a_{t-1,k}}\Phi(a_{t-1,k}, a_{t,k})\overrightarrow{\phi}_t(a_{t-1,k})\overrightarrow{\phi}_f(a_{t-1,k})\overleftarrow{\phi}_f(a_{t-1,k})\upsilon_{t,k}(a_{t-1,k}). \tag{5.22}
$$

The other three messages are defined similarly, and the edge messages are set to uniform. We also define the incoming message from the association variables to the observation:

$$
q_{t,k}(a_{t,k}) = \overrightarrow{\phi}_t(a_{t,k})\overleftarrow{\phi}_t(a_{t,k})\overrightarrow{\phi}_f(a_{t,k})\overleftarrow{\phi}_f(a_{t,k}). \tag{5.23}
$$

This modifies the incoming message (5.15) from the observation factor to the DOA variable $d_t(j)$ as follows:

$$
\upsilon_{t,k}(d_t(j)) = T_{t,k}(d_t(j)) + \frac{\sum_{\ell\neq j} q_{t,k}(\ell)\rho_{t,k}(\ell)}{q_{t,k}(j)} \tag{5.24}
$$

Compared to (5.15), the second constant additive term now measures the activity of the other speakers in the current TF bin based on both $\rho_{t,k}(j)$ that measures the association based on the current speaker DOA estimation, and $q_{t,k}(j)$ that measures the association based on the information from neighbor TF bins. The final inference of the DOA variables remains unchanged (5.16a), and the inference of the associations variable (5.16b) is modified to include also the MRF messages:

$$P(a_{t,k}|\mathbf{z}) \propto \overrightarrow{\phi}_t(a_{t,k})\overleftarrow{\phi}_t(a_{t,k})\overrightarrow{\phi}_f(a_{t,k})\overleftarrow{\phi}_f(a_{t,k})\upsilon_{t,k}(a_{t,t}). \tag{5.25}$$

5.2.7 Complexity and computation time

The complexity of the proposed algorithm depends on the number of microphones (N), number of DOA candidates (M), number of frequencies (K), number of time-frames (T), number of speakers (J) and number of the LBP iterations (denoted as N_{iter}). The algorithm is implemented in two stages. In the first, we calculate the likelihood ratio test (LRT) function $T_{t,k}(m)$ as described in Algorithm 1. Then, we run LBP inference procedure from Algorithm 2.

The calculation of $T_{t,k}(m)$ consist of:

1. Calculate the MVDR-BF: K times $N \times N$ matrix inversion and $K \cdot M$ multiplication of $N \times N$ matrix with $N \times 1$ vector, multiply the results with $N \times 1$ vector, and K scalar divisions - $\mathcal{O}(K \cdot N^3 + K \cdot M \cdot N^2 + K)$.
2. Apply the MVDR-BF on the signal: $T \cdot K \cdot M$ dot products of two $N \times 1$ vectors - $\mathcal{O}(T \cdot K \cdot M \cdot N)$.
3. Calculate the residual noise: Already calculated for the MVDR-BF.
4. Calculate the LRT: $\mathcal{O}(T \cdot K \cdot M)$ operations.

In total the order of magnitude of the required operations:

$$\mathcal{O}(K \cdot N^3 + K \cdot M \cdot N^2 + T \cdot K \cdot M \cdot N). \tag{5.26}$$

For each iteration in the LBP and for each time-step we have the following computations:

1. Compute the messages Ψ: $J \cdot (K+1)$ times element-wise multiplication of $M \times 1$ vectors. Multiply the results with $M \times M$ matrix - $\mathcal{O}(J \cdot K \cdot M + J \cdot M^2)$.

2. Compute $\tilde{\delta}_{t,k,j}(\cdot)$: $J \cdot (K+1)$ times element-wise multiplication of $M \times 1$ vectors - $\mathcal{O}(J \cdot K \cdot M)$.

3. Compute $\rho_{t,k}(\cdot)$: $K \cdot J$ dot product of two $M \times 1$ vectors - $\mathcal{O}(K \cdot J \cdot M)$

4. Compute $\upsilon_{t,k}(\cdot)$ for associations: Simple assignment. No computations required.

5. Compute $\upsilon_{t,k}(\cdot)$ for DOAs: $(J-1) \times K$ operations for the sum computation and then $K \cdot J$ additions of this sum to an $M \times 1$ vector - $\mathcal{O}(K \cdot J \cdot M)$.

In total the order of magnitude of the required operations:

$$\mathcal{O}(N_{\text{iter}} \cdot T \cdot (J \cdot K \cdot M + J \cdot M^2)). \tag{5.27}$$

The final inference algorithm consists of $M \cdot J \cdot T$ for (5.17) and $K \cdot J \cdot T$ for (5.18), which is included in the complexity of (5.27). The actual computation time for typical parameters, is reported in the experimental section 5.3.4.

5.3 Experimental Study

The proposed algorithm was evaluated using both simulated time-varying scenes and real recordings carried out at the Bar-Ilan university (BIU) acoustic lab.

5.3.1 Parameters, evaluation methods and baseline algorithm

In our experiments we used a linear array, therefore the TDOA in (2.2) can be calculated in advance from the predefined grid of DOA candidates and the array constellation. Assuming that the sources are located far from the array (far-field condition), the TDOA in (2.2) is given by $\tau_{m,n} = \frac{1}{c_s} \cdot (r_n \cos(\vartheta_m))$, where ϑ_m is the mth candidate DOA, c_s is the sound velocity and r_n is the distance between the nth microphone and the first microphone. Note that we use the far-field assumption to analytically specify the RTF of the candidates, however, in the experiments we show that the proposed algorithm is not restricted to the reverberation-free far-field case, but can rather be applied in reverberant environments.

The parameters used in the implementation of our algorithm are as follows. The signals are sampled at 16 kHz. The STFT frame-length is set to 64 ms with 75% overlap. The grid of possible azimuth angles ranges between -90^0 and 90^0, with resolution of 2°. The noise PSD matrix was estimated in advance using a clean noise recording.

In our experiments, we observed that the optimal HMM parameter α highly depends on the SNR of the experiment. We therefore select the value of α in each experiment to be in the same order of magnitude of $T_{t,k}$, namely:

$$\alpha = \frac{\sum_{t,k}\left(\max_m \log T_{t,k}(m) - \min_m \log T_{t,k}(m)\right)}{T \cdot K}. \tag{5.28}$$

The parameter of the MRF model was set to $\beta = 0.5$, which was selected using a grid search. The LBP algorithm was stopped either after $N_{\max}$ iterations, or when the maximum change in the log messages between subsequent iterations was smaller than $\varepsilon = 10^{-3}$, where $N_{\max} = 20$ or 50, for the simulation and lab experiments, respectively.

We have two options of how to define the initial DOA message $\Omega_j(m)$. The first option is to assume that the initial DOA is known, so in $\Omega_j(m)$ the known initial DOA is assigned with probability one and the other DOAs are assigned with zero probabilities. The second option is to assume that the initial DOA is unknown, to randomly generate the values of $\Omega_j(m)$, and to normalize them so they sum to one. In this option, we avoid using a uniform message since it may cause the estimates to collapse to one track.

In order to assess the performance of the algorithm, we evaluated both the tracking accuracy and the separation results. The tracking estimation error was first evaluated for each speaker using the root mean square error (RMSE) measure, namely $e_d(j) = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(\hat{d}_t(j) - d_t(j))^2}$. The final score is obtained by averaging this value for all speakers. For the separation performance, we used the source to distortion ratio (SDR), source to interference ratio (SIR) and source to artifacts ratio (SAR) scores, evaluated by the BSS-Eval Toolbox [44].

As a baseline method we used the variational-based tracking algorithm proposed in [18]. In this algorithm the covariance matrix of the RTF is a priori defined, and we set it to $\boldsymbol{\Sigma}_a = 10\mathbf{I}$. The transition matrix was defined as in (5.4) with $\alpha = 0$. This algorithm requires the oracle initial DOA of the speakers for the RTF initialization. For fair comparison, we initialized both algorithms with the true DOA, and separately examined the performance of the proposed algorithm also with random initialization. For the same reason, we implemented the same separation procedure using (5.18) for both methods.

In addition, we report the separation results obtained using the oracle DOA in the construction of the MVDR-BF as well as the oracle separation mask, which was computed using the known separated speech signals. It is the best performance that may be achieved with the separation procedure defined in (5.18), and can therefore serve as an upper bound for the performance of the proposed algorithm.

Note that a comparison to the former models from this work is not possible since those models estimate the DOA distribution for each time frame rather than the actual trajectories, and also do not handle the separation task.

5.3.2 Simulation experiment

For the simulated data, clean anechoic speech signals were drawn from the TIMIT database [37]. The speakers were randomly selected from a subset of 26 speakers. Speech utterances of the same speaker were concatenated to obtain a 5 s long speech signal. Note that the proposed method cannot perform well when long silence periods exist, since it stops tracking the speaker whenever he is inactive. However, the proposed method can tolerate small natural silence periods. Therefore, long silence segments were removed, so that all the speakers are almost simultaneously active during the entire signal.

To simulate moving sources, we used the signal generator.[1] The room dimensions

[1] www.audiolabs-erlangen.de/fau/professor/habets/software/signal-generator

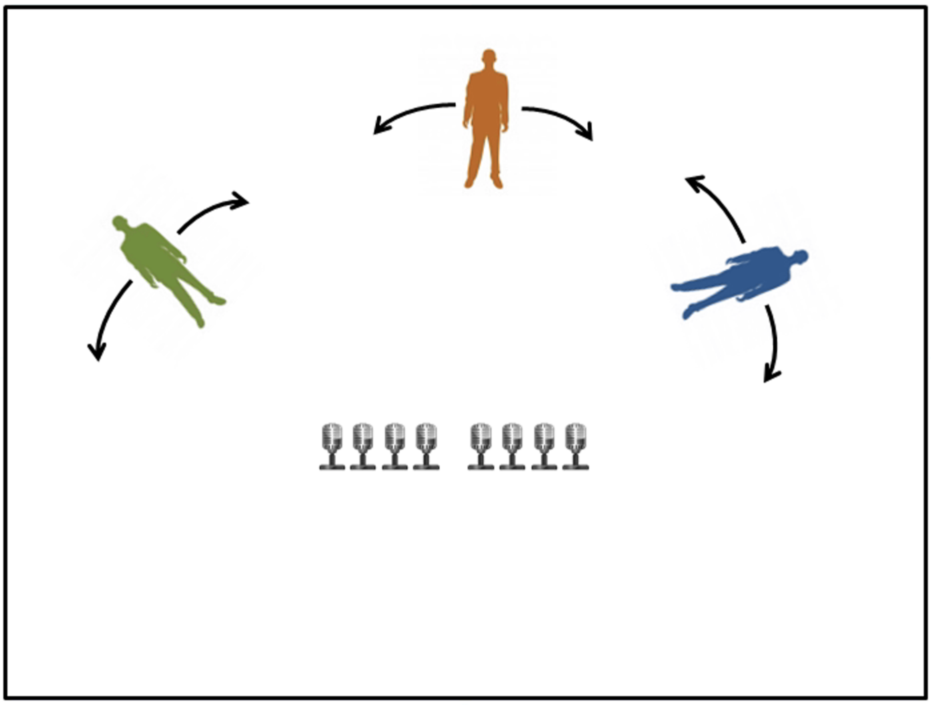


Figure 5.5: An illustration of the simulation setup.

were set to $6 \times 4 \times 3$ m with reverberation time $T_{60} \sim 200$ ms. The signals were captured by an eight-microphone linear array with inter-distances of $[3, 3, 3, 8, 3, 3, 3]$ cm. The array center was positioned in the center of the room, in coordinates $(3, 2, 1)$ m. The measured signals were contaminated by an additive babble diffuse noise with various SNR levels. The diffuse noise sound-field was generated using the noise generator software.[2]

Three moving speakers were simulated, with initial DOAs set to $36°$, $90°$ and $144°$, respectively. The speakers moved from their initial positions along an arc of a circle with a radius of 1 m from the array center. Their time-varying DOA has a sinusoidal form, with time period randomly selected between $1 - 2.5$s, and amplitude also randomly selected between $5° - 8°$. The simulated setup is depicted in Fig. 5.5.

An example of the estimated TF associations as compared with the true associations of one of the speakers is given in Fig. 5.6. For the clarity of the demonstration we focus on a short segment of 2 s. We observe a good match between the true and the estimated associations, indicating that the proposed algorithm successfully recovers the TF activity

[2] www.audiolabs-erlangen.de/fau/professor/habets/software/noise-generators

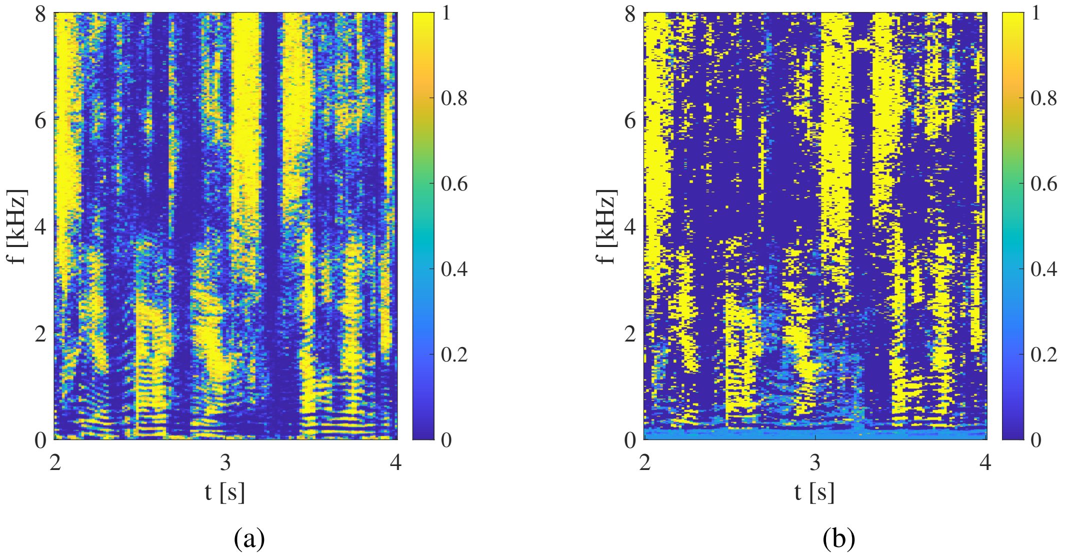


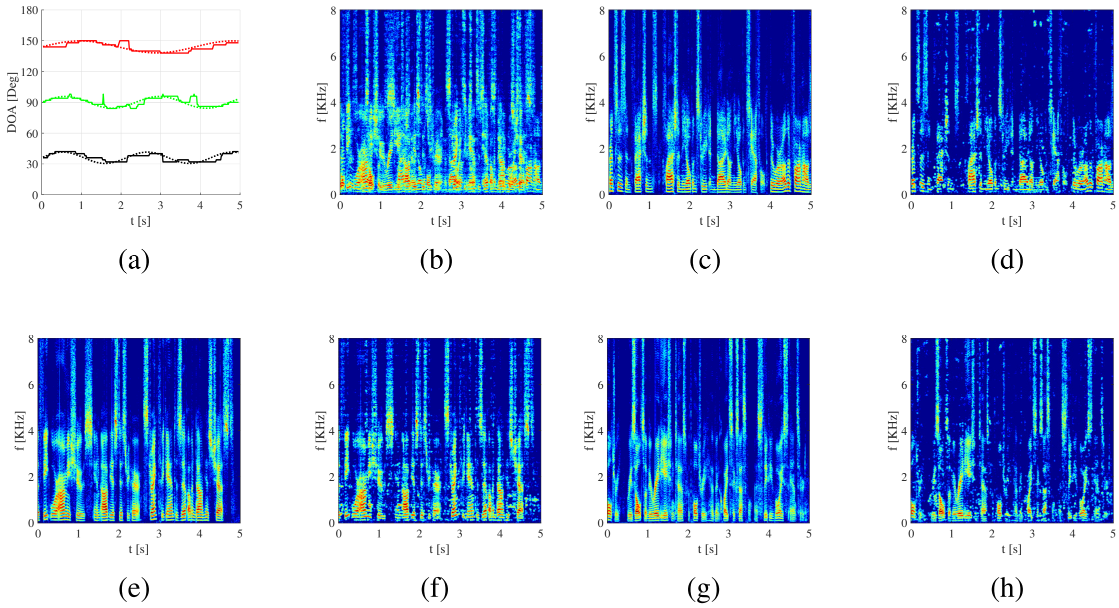
Figure 5.6: Comparison of TF associations of the first speaker in the simulation experiment. The ground truth (left) and the estimated associations (right) are depicted.

of the speakers.

An example of the DOA estimation and the separation results obtained by the proposed algorithm is illustrated in Fig. 5.7. It can be seen that the proposed algorithm successfully recovers the trajectory of all speakers. True and estimated spectrograms of all the speakers are also depicted, demonstrating good separation performance.

The tracking and separation results were evaluated on 200 Monte-Carlo (MC) trials with different speakers and different trajectories for 3 SNR levels: 5 dB, 10 dB and 25 dB. The statistics of the obtained scores are reported in boxplots in the left column of Fig. 5.8 with outliers omitted for clarity. It can be seen that for the proposed algorithm the results of the uniform and the MRF models are comparable, and that they outperform the reference algorithm [18] on both tracking and separation tasks.

In addition, we examined the performance of the proposed method with respect to different room environments. Here, we fixed the SNR to 25 dB, and examined three reverberation times: 200 ms, 400 ms and 600 ms, and two source distances with respect to the center of the array: 1 m and 1.5 m. The tracking and separation results were averaged over 100 MC trials with different speakers and different trajectories. The results of this



(a) (b) (c) (d)

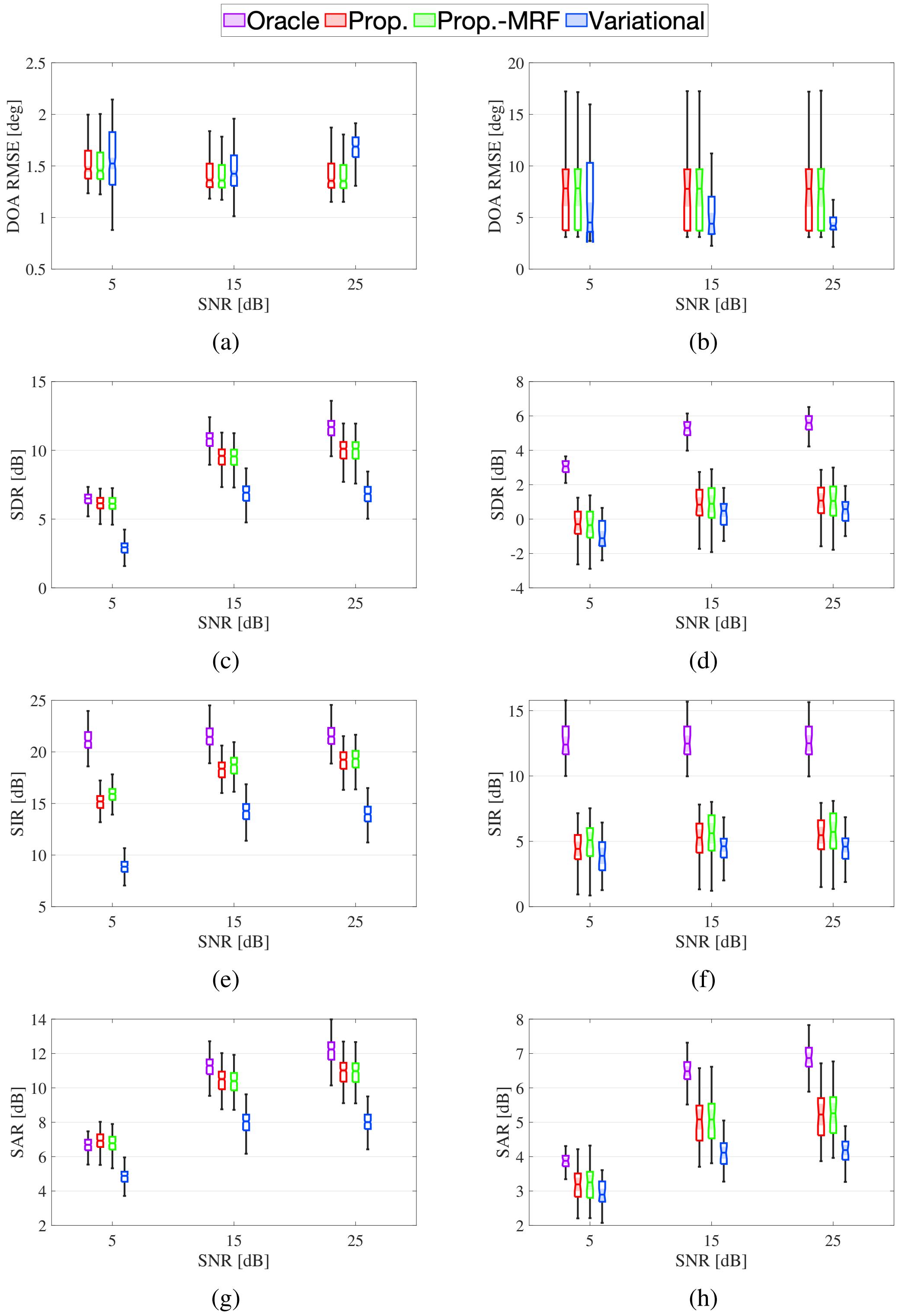
(e) (f) (g) (h)

Figure 5.7: An example of the simulation results. True DOA in dashed line and estimated DOA in solid line (a), first microphone mixed signal spectrogram (b), clean and estimated spectrogram of the first speaker (c+d), the second speaker (e+f) and the third speaker (g+h).

experiment are reported in Fig. 5.9. We observe a decrease in the separation scores and an increase in the DOA RMSE for higher reverberation levels or larger source-microphone distance. The difference in the performance between 1 m and 1.5 m distance becomes more significant for higher reverberation levels, apparently due to the fact that in high reverberation the direct-to-reverberant power ratio becomes much lower as the source-microphone distance increases.

5.3.3 Laboratory experiment

In addition to the simulated experiment, we evaluated the proposed algorithm using real recordings carried out at the BIU acoustic lab. We first defined two limited arcs on a circle with radius of ~ 2 m: the first arc between $20°$-$75°$ and the other between $120°$-$165°$. Seven speakers participated in our experiment, five males and two females. Each speaker moved back and forth while speaking with a natural random trajectory on each of the defined arcs. The length of each recording was approximately 30s. The signals were captured by an eight-microphone linear array with inter-distances of $[3, 3, 3, 6, 3, 3, 3]$ cm.



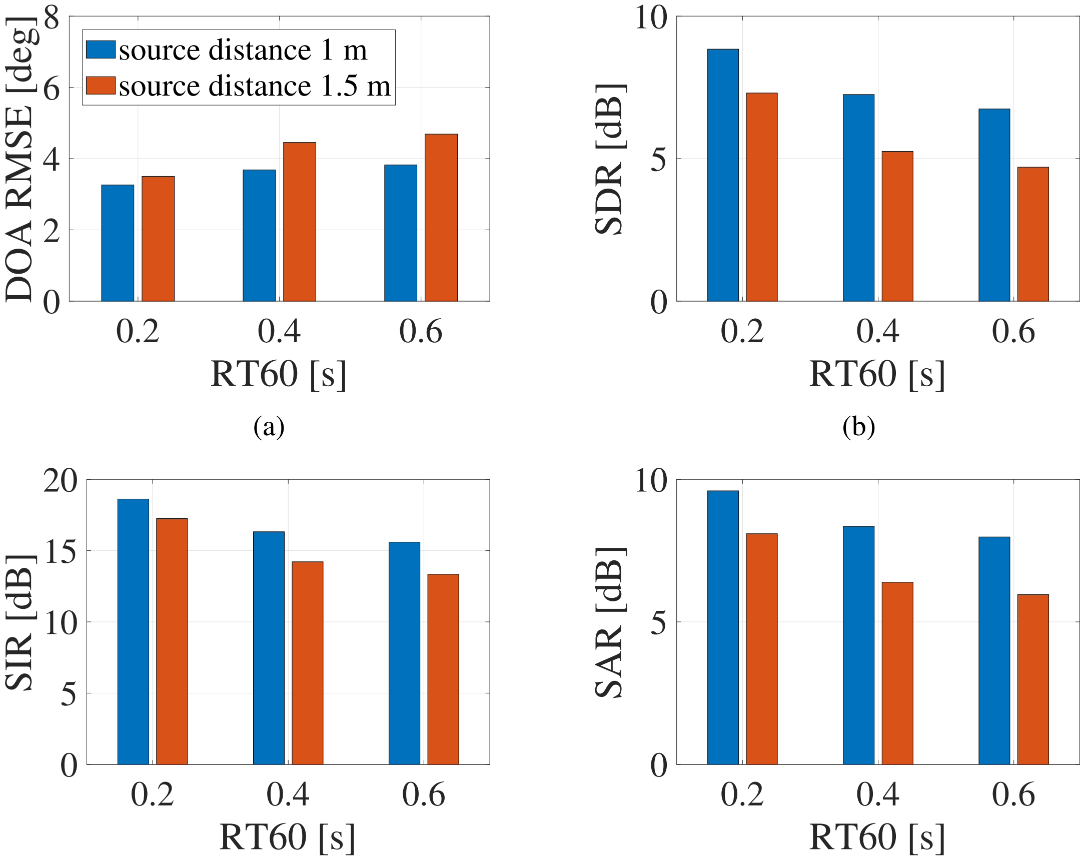
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 5.8: Simulation and lab experiment: separation and tracking performance measures for various SNRs for simulation (left) and lab experiments (right). The results are reported for the reference variational method [18] and for the two versions of the proposed method, with the simple uniform prior of the associations (Prop.) and with the more complex MRF-model as described in Section 5.2.6 (Prop.-MRF). In addition, we report the separation results obtained using the oracle DOA in the construction of the MVDR-BF as well as the oracle separation mask, which was computed using the known separated speech signals.



(a)

(b)

(c)

(d)

Figure 5.9: Separation and tracking performance measures for three reverberation times: 200 ms, 400 ms and 600 ms, and two source distances with respect to the center of the array: 1 m and 1.5 m, averaged over 100 MC trials, with SNR$=25$ dB.

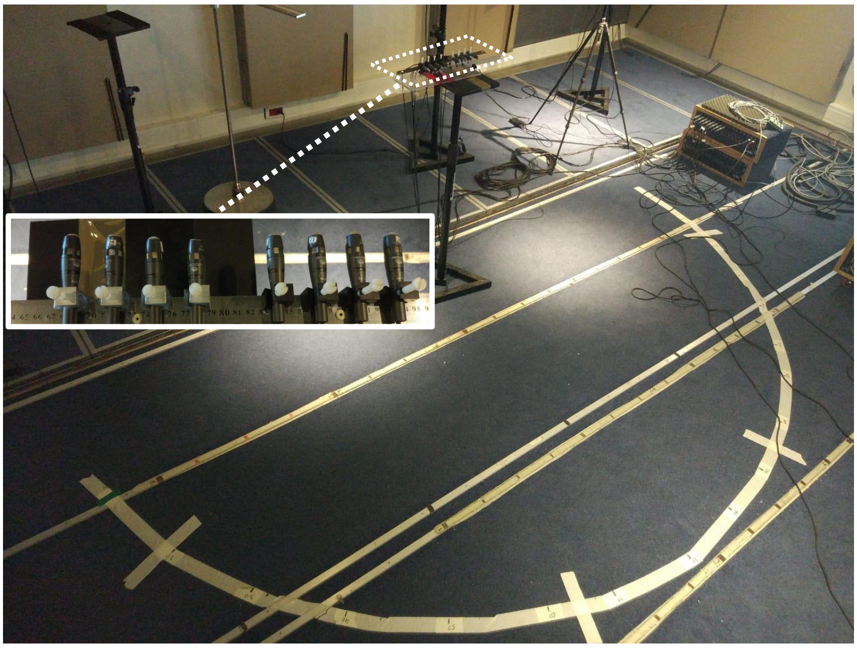


Figure 5.10: A photo of the experimental setup at the BIU lab.

The array was located in the center of the designated circle, in a distance of approximately 1.5 meters from one of the walls. A photograph of the room configuration is given in Fig. 5.10. The reverberation time was set to $T_{60} \sim 450$ms by adjusting the controllable room panels. A diffuse babble noise was also separately recorded by the same array using 4 loudspeakers facing the room corners. Finally, after discarding few utterances due to technical problems in the recordings, we generated 29 combinations of different pairs of speakers with noise added with different SNR levels.

In order to evaluate the results we need both the clean speech for the separation evaluation, and the ground-truth trajectory for the tracking evaluation. For the separation evaluation we used the separately recorded speech signals in the first microphone as a reference. For the ground-truth DOA of the speakers we used Marvelmind indoor navigation system.[3] This system consists of a single mobile device and four stationary devices. The coordinates of the mobile device are reported w.r.t. the stationary devices with reported measurement error of ±2 cm. In practice, we observed that occasionally this device introduces small glitches, apparently due to noise or measurement instability. In the beginning

[3] `https://marvelmind.com/product/starter-set-hw-v4-9/`

of our experiment, we measured the microphone locations, and then each participant held the mobile device during his recording session. The ground-truth DOA is computed as the angle between the microphone array and the line connecting the center of the array and the speaker location.

An example of the DOA estimation obtained by the proposed method with random DOA initialization is shown in Fig. 5.11 (a). The estimated trajectory is close to the ground truth trajectory as measured by the indoor navigation system. Note that although the estimated DOAs of one of the speakers deviates from the true trajectory around $t = 25$s, the algorithm successfully traces back the true trajectory after few seconds. Figure 5.11 (b) shows an example of the DOA estimation obtained with random DOA initialization for a case with three speakers that two of them have close trajectories. It can be seen that the proposed algorithm successfully tracks the three speakers for almost the entire signal duration. The estimated trajectories deviate from the ground truth at the end of the signal when two speakers get closer to each other.

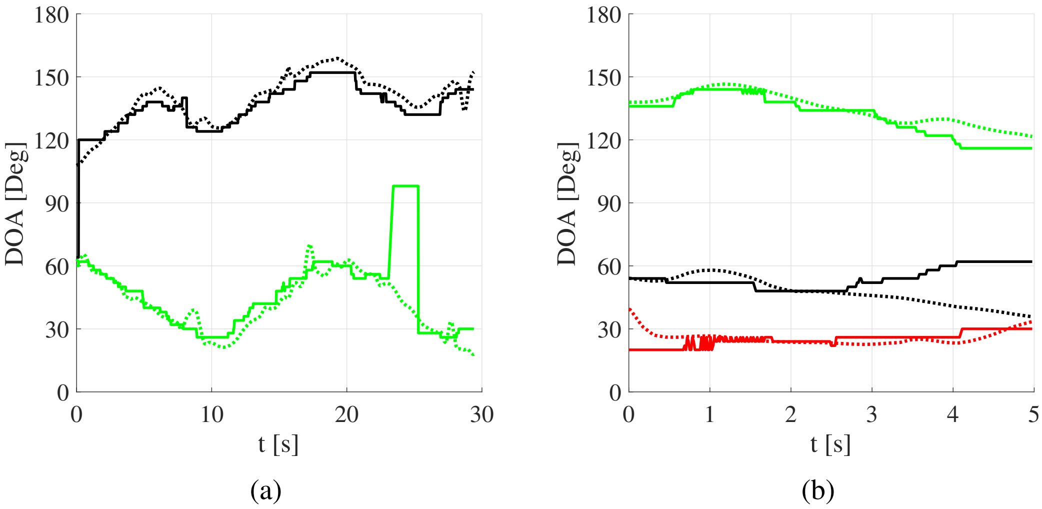


Figure 5.11: Examples of the tracking results in the lab experiment: two distant speakers in a full 30 s recording (left) and three speakers, two of them very close to each other, in a segment of 5 s recording (right). Dashed and solid lines correspond to ground truth (obtained by the indoor navigation system) and estimated trajectories, respectively. The initial DOAs were set randomly. In the three speakers case the estimated trajectories deviate from the ground truth at the end of the signal when two speakers get closer to each other.

The statistics of the 29 different 2-speakers scenarios are reported in boxplots in the right column of Fig. 5.8. While the proposed algorithm outperforms the reference algorithm in the separation task for all SNR values, in the tracking task it obtains higher errors. Comparing the uniform and the MRF models in the proposed algorithm, we observe a slight advantage to the latter in terms of the SIR measure as reflected from the median and the 75 percentile. This advantage is more pronounced in the 5 dB SNR case. Note that the DOA-RMSE might be biased due to measurement errors in the ground-truth DOA, as mentioned above. Note also that the ground-truth separated speech signals, taken as the measurements of the first microphone, cannot serve as a perfect reference as well, which may explain the relatively low separation scores. For subjective evaluation, the reader is referred to our website.[4]

We also examined the sensitivity of the proposed algorithm to the DOA initialization. A comparison of the DOA RMSE obtained by the proposed algorithm with either ground truth or random initialization is given in Fig. 5.12. It is observed that the error is increased by approximately 1 degree for most of the readings. This small increase in the error indicates that the proposed algorithm can track the speakers without prior knowledge on their initial position.

We also examined the dependency of separation quality measures on the gender of the speakers. We compared mixtures of same gender speakers, i.e. male and female, with mixtures of male and female speakers. Analyzing the results, did not show any significant differences. This conclusion might need further investigations, as the number of examples is small.

5.3.4 Computation time

In this section, we report the average computation time of each iteration and the performance of the proposed algorithm and the baseline algorithm as a function of the number of iterations for the simulation experiment. The computation time was calculated using

[4] `http://www.eng.biu.ac.il/gannot/speech-enhancement/`

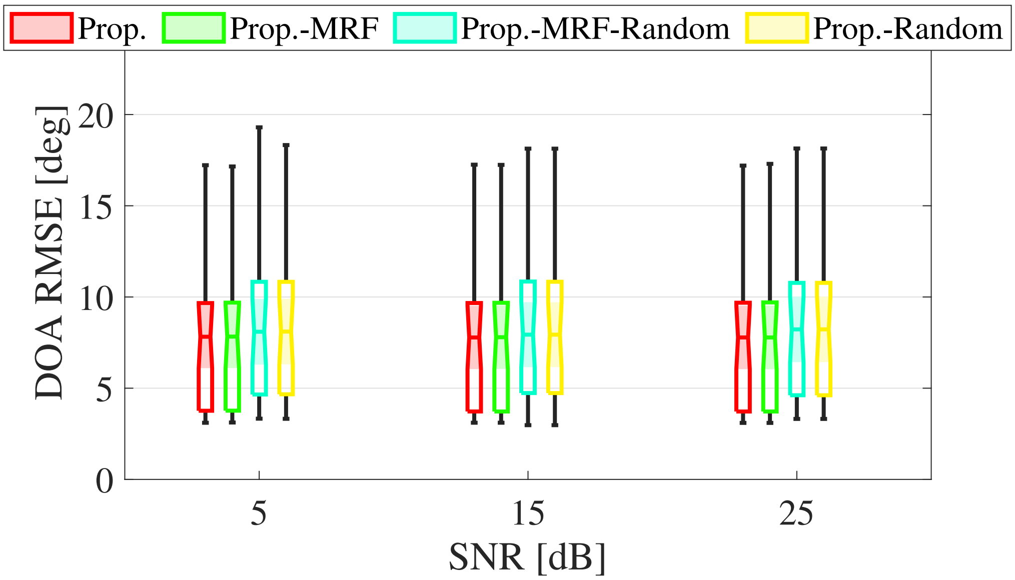
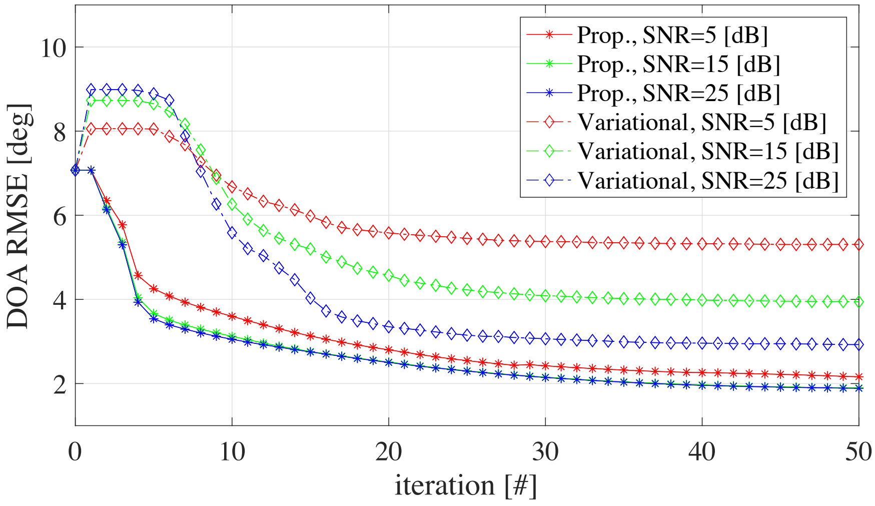
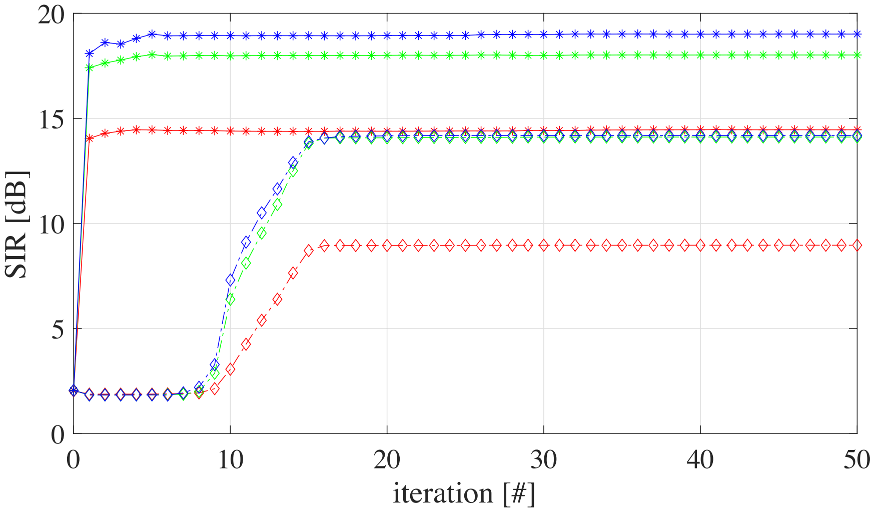


Figure 5.12: Comparing DOA estimation performance with random and oracle initialization.

2.3 GHz Intel Core i9 single CPU, with 16 GB 2400 MHz DDR4 memory. The algorithm was implemented using Matlab©, without using the parallel computing utility. In our experiments, the recording length was 30 seconds. The parameters were: $N = 8$, $K = 513$, $J = 3$, $M = 91$, $T = 309$ and $N_{\text{max}} = 50$. The average computation time was roughly 3.8 s per iteration per second of input signal, compared to an average of 6.6 s for the reference algorithm. Note also that the total computation time linearly depends on the number of iterations. In Fig. 5.13, we report the tracking and separation performance measures as a function of the number of iterations. It is demonstrated that in terms of the separation performance, the proposed algorithm converges within 5 iterations, compared to 15 iterations required by the reference algorithm, and also obtains better SIR scores after convergence. For the DOA estimation, the proposed algorithm converges after 35 iterations to a lower RMSE compared to that achieved by the reference algorithm, which converges after 20 iterations. Note also that the DOA RMSE obtained by the proposed algorithm decreases to $3^\circ - 4^\circ$, already within 5 iterations. Therefore, when the available computation time is limited, we can run only 5 iterations of the proposed algorithm to obtain maximal separation performance and low DOA RMSE of less than 5°.



(a)



(b)

Figure 5.13: Tracking and Separation performance of a particular scenario as function of the number of iterations.

Chapter 6

Conclusions

In this chapter we summarize our work and outline some future research directions.

6.1 Research summary

We have presented three algorithms for simultaneous tracking, separation and noise reduction of multiple speakers.

First we stated a statistical model for the observations given both the DOA trajectories and the TF bin associations and simplify the obtained conditional distribution. Then we used statistical inference method to infer on the hidden data which is defined to be the DOA trajectories and the TF bin associations. We used three different models for the hidden data.

The first model is the instantaneous model, where the hidden data is defined to be the DOA association of each TF bin, which is actually the DOA of the active speaker in this bin. We modeled the hidden data independently over time, with shared prior distribution over the candidates DOA. For static scenario, this prior is constant and therefore we proposed to use batch-EM algorithm for inference. We further derived a recursive-EM algorithm for the dynamic case, where those priors are changing over time smoothly.

The second model has two variations, the parallel-HMM and the coupled-HMM. In the the parallel-HMM, the hidden data was defined similarly to the first model, however, a time dependence was introduced, by modeling each frequency band as a Markov chain, with transition matrix that allows for small changes in the DOA. The inference is done using standard FB algorithm. In the second variation, per-frequency dependence was also introduced, and we used an extended FB algorithm for inference.

In the third model the hidden dat defined to be both the DOA trajectories and the associations. The posterior of the hidden data given the observations was modeled as a factor graph. We used the LBP inference algorithm, to derive a novel inference scheme where both the DOA trajectory and the separation mask are jointly obtained.

For each model, we evaluated the performance using an experimental study on both simulated data and real-life recordings, and we demonstrated the advantage of the proposed algorithm compared to reference methods.

6.2 Topics for Further Research

The present work can be extended in the following directions:

1. **Improve the association model**: In the proposed FG model, we used MRF model as a prior for the association of the TF bins to the speakers. However, the improvement in the performance was negligible. Future work could replace this model with a more accurate model. One possible option is to train a neural network model on clean speech signals to learn the probability of a speaker to be active at each frequency bin, given the previous time step speech activity. This neural network can then be used as a prior for the associations.

2. **Add an option to associate the TF bins to noise**: Currently we assign each TF bin to one of the speakers, however, we observed empirically that many of the bins do not include speech component, but are dominated by noise. Previous works (for example [8]) proposed to add a non-speaker candidate to improve the separation results. Applying a similar approach in our model may also improve the separation

results. Furthermore, it can be utilized to estimate the noise covariance matrix, which currently assumed to be known in advance.

3. **Learn the hyper-parameters using the EM algorithm**: In both the HMM and the FG models, we used Markov chains to model smooth transition of the DOA over time. The transition matrices for those chains were defined in advance, using heuristic methods to obtain best performance for our tasks. A future direction might be, to use the EM algorithm to learn these matrices during the algorithm application, as done for classic HMM models [38]. The proposed inference scheme already produces a soft associations for the hidden variables which may serve as the E-step. In the M-step one should find the MLE of the unknown parameters given those associations.

4. **Extend the algorithm to 2D tracking**: Similar to [10], several microphone arrays can be used to obtain a full 2D tracking, rather than DOA tracking. The straightforward approach for this extension is to substitute the DOA candidates, by a 2D grid of candidates, and to modify the transition matrix accordingly, which may be intractable, due to large number of candidates. A better approach for future study is to use same scheme for each microphone array to obtain per-array decisions, and then optimally combine these decisions, while keeping the entire 2D trajectory smooth.

Appendix A

Factor Graphs

In this section, we briefly review the definition of factor graphs and their inference methods based on [27; 45].

A.1 Definition

Let $\{x_1, x_2, ..., x_Q\}$ be a set of Q discrete-valued random variables. We consider the joint probability mass function $P(\mathbf{x}) = P(x_1, x_2, ..., x_Q)$, which is assumed to be factored into a product of functions:

$$P(\mathbf{x}) = \frac{1}{C} \prod_{u \in \mathcal{U}} f_u(\mathbf{x}_u) \tag{A.1}$$

where u is an index that labels the functions from a set $\mathcal{U}$, where each function $f_u(\mathbf{x}_u)$ has arguments $\mathbf{x}_u \subset \{x_1, x_2, ..., x_Q\}$. We assume that the functions $f_u(\mathbf{x}_u)$ are non-negative and finite, so that $P(\mathbf{x})$ is a well-defined probability distribution. Here, C is a normalization constant.

A factor graph is a bipartite graph that expresses the factorization structure in (A.1). A factor graph has a variable node (which we draw as a circle) for each variable x_i, and a factor node (which we draw as a square) for each function f_u, with an edge connecting variable node x_i to factor node u if and only if $x_i \in \mathbf{x}_u$.

A.2 Inference

For a given graph with given factors, one may be interested in two different goals. The first is to find the marginals of each variable, i.e. $P(x_i)\ \forall i$, and the other is to find the most probable state, i.e. $\mathrm{argmax}_{\mathbf{x}}\, P(\mathbf{x})$. An exact inference for factor graphs is obtained using the belief propagation (BP) algorithm. When implemented for computing the marginal p.d.f., the BP algorithm is also known as the *sum-product* algorithm, and when implemented for finding the most probable state, it is called the *max-product* algorithm. In the sum-product algorithm messages are sent from the factors to the variables and vice-verse, using the following equations:

$$n_{i\to u}(x_i) = \prod_{c\in\mathcal{G}\{x_i\}/u} m_{c\to i}(x_i) \tag{A.2a}$$

$$m_{u\to i}(x_i) = \sum_{\mathbf{x}_u/x_i} f_u(\mathbf{x}_u) \prod_{j\in\mathcal{G}\{u\}/x_i} n_{j\to u}(x_j) \tag{A.2b}$$

where $n_{i\to u}(x_i)$ is the message from the ith variable to the uth factor, $m_{u\to i}(x_i)$ is the opposite direction message, $\mathcal{G}\{x_i\}$ is the set of neighbouring factors of x_i and $\mathcal{G}\{u\}$ is the set of neighbouring variables of u. We can then obtain the marginal probability of a particular variable x_i using:

$$P(x_i) \propto \prod_{u\in\mathcal{G}\{x_i\}} m_{u\to i}(x_i) \tag{A.3}$$

where the sign $\propto$ means that one should normalize this expression to obtain the final distribution. In the *max-product* algorithm summations are replaced by the `max` operator. The max-product algorithm is out of the scope of this article.

The sum-product algorithm is proved to converge to the true marginals in tree-structured graphs [38]. However, when the graph contains loops this algorithm is not proved to coverage to the true marginal. The loopy belief propagation (LBP) [42] is an extension of the BP algorithm for loopy graphs, in which messages are updated repeatedly, in an arbitrary order, until a termination condition is met. In practice, it has been observed that this algorithm often provides good estimates of the marginals.

An alternative approximate inference is the Gibbs sampling method. In this method, we first randomly initialize all hidden variables to a value from their range. Next, we iterate over all variables, and sample from their conditional distribution given all other variables. Due to the factored joint distribution, this conditional distribution is given by:

$$P(x_i|\mathbf{x}/x_i) \propto \prod_{\{a|x_i \in x_a\}} f_a(x_a). \tag{A.4}$$

After N iterations, we get sequence of $[x_i^{(1)} \dots x_i^{(N)}]$ values for each variable. Finally, the marginal distribution for each variable is given by:

$$P(x_i = a) = \frac{\sum_{n=1}^{N} \mathbb{1}_{\{x_i^{(n)}=a\}}}{N} \tag{A.5}$$

For $N \to \infty$ this estimate converges to the true marginal, however, for finite N, it is common to ignore some number of samples at the beginning (the so-called *burn-in period*) in order to improve the accuracy of the algorithm.

Appendix B

Derivation of the Messages from the Observation Factors

In this section, we derive the messages from the observation factors to its neighboring variables. For general derivation, we assume here that each variable sends a message to the observations. We denote by $\delta_{t,k,j}(:)$ and $q_{t,k}(:)$ the messages from the DOA and the association variables, respectively.

B.1 The message from the observations to the association variables

Using (A.2b) the messages to $a_{t,k}$ are given by:

$$\upsilon_{t,k}(a_{t,k}) = \sum_{d_t(1)} \sum_{d_t(2)} \dots \sum_{d_t(J)} \Upsilon_{t,k}(a_{t,k}, d_t(1) \dots d_t(J)) \prod_{i=1}^{J} \delta_{t,k,i}(d_t(i)).$$

Substituting the definition of $\Upsilon_{t,k}$ (5.7) we obtain:

$$\upsilon_{t,k}(a_{t,k}) = \sum_{d_t(1)} \sum_{d_t(2)} \dots \sum_{d_t(J)} T_{t,k}(d_t(a_{t,k})) \prod_{i=1}^{J} \delta_{t,k,i}(d_t(i)). \tag{B.1}$$

Note that the expression $T_{t,k}(d_t(a_{t,k}))$ is constant for all summations except for the sum over $d_t(a_{t,k})$, hence we rearragne the summations as follows:

$$\upsilon_{t,k}(a_{t,k}) = \sum_{d_t(a_{t,k})} T_{t,k}(d_t(a_{t,k})) \sum_{d_t(:)/d_t(a_{t,k})} \prod_{i=1}^{J} \delta_{t,k,i}(d_t(i)).$$

Since each message $\delta_{t,k,i}(d_t(i))$ is influenced by only one summation, we can switch the sum and product operations:

$$\sum_{d_t(a_{t,k})} T_{t,k}(d_t(a_{t,k})) \cdot \delta_{t,k,a_{t,k}}(d_t(a_{t,k})) \prod_{i \neq a_{t,k}} \sum_{d_t(i)} \delta_{t,k,i}(d_t(i)).$$

In order to further simplify this expression, we multiply and divide it by the term $\sum_{d_t(a_{t,k})} \delta_{t,k,a_{t,k}}(d_t(a_{t,k}))$ to obtain

$$\frac{\sum_{d_t(a_{t,k})} T_{t,k}(d_t(a_{t,k})) \cdot \delta_{t,k,a_{t,k}}(d_t(a_{t,k}))}{\sum_{d_t(a_{t,k})} \delta_{t,k,a_{t,k}}(d_t(a_{t,k}))} \underbrace{\prod_{i} \sum_{d_t(i)} \delta_{t,k,i}(d_t(i))}_{\text{Const}}. \tag{B.2}$$

Since the messages are not normalized anyway, we can ignore the constant term, and we finally obtain:

$$\upsilon_{t,k}(a_{t,k}) \propto \frac{\sum_m T_{t,k}(m) \cdot \delta_{t,k,a_{t,k}}(m)}{\sum_m \delta_{t,k,a_{t,k}}(m)} \equiv \rho_{t,k}(a_{t,k}) \tag{B.3}$$

B.2 The messages from the observations to the DOA variables

The incoming messages are coming from $d_t(1), \ldots, d_t(j-1), d_t(j+1), \ldots, d_t(J)$ and $a_{t,k}$, therefore:

$$\upsilon_{t,k}(d_t(j)) = \sum_{a_{t,k}} \sum_{d_t(:)/d_t(j)} T_{t,k}(d_t(a_{t,k})) q_{t,k}(a_{t,k}) \prod_{i \neq j} \delta_{t,k,i}(d_t(i))$$

where $q_{t,k}(a_{t,k})$ is uniform for the uniform distribution model (5.5) or defined by (5.23) for the MRF model (5.19). We split the first summation over all possible values of $a_{t,k} \in [1 \dots J]$ to a sum over j and summations over all other values:

$$= \underbrace{\sum_{d_t(:)/d_t(j)} T_{t,k}(d_t(j)) q_{t,k}(j) \prod_{i \neq j} \delta_{t,k,i}(d_t(i))}_{\text{(*)}} + \sum_{a_{t,k} \neq j} q_{t,k}(a_{t,k}) \underbrace{\sum_{d_t(:)/d_t(j)} T_{t,k}(d_t(a_{t,k})) \prod_{i \neq j} \delta_{t,k,i}(d_t(i))}_{\text{(**)}}\,.$$

This expression consists of two terms. In $(*)$ the term $T_{t,k}(d_t(j))q_{t,k}(j)$ depends on $d_t(j)$, hence we take it out of the summation and switch the order of the sum and product operations to obtain:

$$(*) = T_{t,k}(d_t(j)) q_{t,k}(j) \prod_{i \neq j} \sum_{d_t(i)} \delta_{t,k,i}(d_t(i))$$

The term $(**)$ is same as (B.1) and similarly to (B.2) it can simplified to:

$$(**) = \rho_{t,k}(a_{t,k}) \prod_{i \neq j} \sum_{d_t(i)} \delta_{t,k,i}(d_t(i)).$$

The overall message is now given by:

$$= T_{t,k}(d_t(j)) q_{t,k}(j) \underbrace{\prod_{i \neq j} \sum_{d_t(i)} \delta_{t,k,i}(d_t(i))}_{\text{const}} + \sum_{a_{t,k} \neq j} \rho_{t,k}(a_{t,k}) q_{t,k}(a_{t,k}) \underbrace{\prod_{i \neq j} \sum_{d_t(i)} \delta_{t,k,i}(d_t(i))}_{\text{const}}\,.$$

Dividing the message by the constant $q_{t,k}(j) \prod_{i \neq j} \sum_{d_t(i)} \delta_{t,k,i}(d_t(i))$, we finally obtain:

$$m_{t,k}(d_t(j)) \propto T_{t,k}(d_t(j)) + \frac{\sum_{a_{t,k} \neq j} q_{t,k}(a_{t,k}) \rho_{t,k}(a_{t,k})}{q_{t,k}(j)}. \tag{B.4}$$

Bibliography

[1] J. DiBiase, H. Silverman, and M. Brandstein, *Microphone arrays : signal processing techniques and applications*. Springer Verlag, 2001, ch. Robust Localization in Reverberant Rooms, pp. 157–180.

[2] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[3] A. Hyvärinen and E. Oja, “Independent component analysis: Algorithms and applications,” *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, May 2000. [Online]. Available: http://dx.doi.org/10.1016/S0893-6080(00)00026-5

[4] O. Yilmaz and S. Rickard, “Blind separation of speech mixtures via time-frequency masking,” *IEEE Trans. Sign. Process.*, vol. 52, no. 7, pp. 1830–1847, 2004.

[5] S. Gannot, E. Vincent, S. Markovich-Golan, A. Ozerov, S. Gannot, E. Vincent, S. Markovich-Golan, and A. Ozerov, “A consolidated perspective on multimicrophone speech enhancement and source separation,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 4, pp. 692–730, 2017.

[6] E. Vincenet, T. Virtanen, and S. Gannot, *Audio Source Separation and Speech Enhancement*. Wiley, 2018.

[7] S. Makino, *Audio Source Separation*. Springer, 2018.

[8] M. I. Mandel, R. J. Weiss, and D. P. Ellis, “Model-based expectation-maximization source separation and localization,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 18, no. 2, pp. 382–394, 2010.

[9] M. I. Mandel and N. Roman, “Enforcing consistency in spectral masks using markov random fields,” in *European Signal Processing Conference (EUSIPCO)*, 2015, pp. 2028–2032.

[10] O. Schwartz and S. Gannot, “Speaker tracking using recursive EM algorithms,” *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 2, pp. 392–402, 2014.

[11] M. D. Titterington, “Recursive parameter estimation using incomplete data,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 257–267, 1984.

[12] O. Cappé and E. Moulines, “On-line expectation–maximization algorithm for latent data models,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 71, no. 3, pp. 593–613, 2009.

[13] J. Traa and P. Smaragdis, "Multichannel source separation and tracking with RANSAC and directional statistics," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 12, pp. 2233–2243, 2014.

[14] O. Schwartz, Y. Dorfan, E. A. Habets, and S. Gannot, "Multi-speaker DOA estimation in reverberation conditions using expectation-maximization," in *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2016.

[15] Y. Dorfan, O. Schwartz, B. Schwartz, E. A. Habets, and S. Gannot, "Multiple DOA estimation and blind source separation using estimation-maximization," in *IEEE International Conference on the Science of Electrical Engineering (ICSEE)*, 2016.

[16] O. Schwartz, Y. Dorfan, M. Taseska, E. A. Habets, and S. Gannot, "DOA estimation in noisy environment with unknown noise power using the EM algorithm," in *Hands-free Speech Communications and Microphone Arrays (HSCMA)*, 2017, pp. 86–90.

[17] N. Roman and D. Wang, "Binaural tracking of multiple moving sources," *IEEE Trans. Audio Speech Lang. Process.*, vol. 16, no. 4, pp. 728–739, 2008.

[18] T. Higuchi, N. Takamune, T. Nakamura, and H. Kameoka, "Underdetermined blind separation and tracking of moving sources based ondoa-hmm," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 3191–3195.

[19] D. Kounades-Bastian, L. Girin, X. Alameda-Pineda, S. Gannot, and R. Horaud, "A variational em algorithm for the separation of time-varying convolutive audio mixtures," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 24, no. 8, pp. 1408–1423, 2016.

[20] M. Swartling, N. Grbic, and I. Claesson, "Direction of arrival estimation for multiple speakers using time-frequency orthogonal signal separation," in *IEEE International Conference on Acoustics Speech and Signal Processing Proceedings (ICASSP)*, vol. 4, 2006, pp. 833–836.

[21] A. Brendel, B. Laufer-Goldshtein, S. Gannot, R. Talmon, and W. Kellermann, "Localization of an unknown number of speakers in adverse acoustic conditions using reliability information and diarization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 7898–7902.

[22] K. Weisberg, S. Gannot, and O. Schwartz, "An online multiple-speaker doa tracking using the Cappé-Moulines recursive expectation-maximization algorithm," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 656–660.

[23] K. Weisberg and S. Gannot, "Multiple speaker tracking using coupled HMM in the STFT domain," in *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, Le Gosier in Guadeloupe, French West Indies, Dec. 2019.

[24] K. Weisberg, B. Laufer-Goldshtein, and S. Gannot, "Simultaneous tracking and separation of multiple sources using factor graph model," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2848–2864, 2020.

[25] L. K. Saul and M. I. Jordan, "Mixed memory Markov models: Decomposing complex stochastic processes as mixtures of simpler ones," *Machine learning*, vol. 37, no. 1, pp. 75–87, 1999.

[26] S. Zhong and J. Ghosh, "HMMs and coupled HMMs for multi-channel EEG classification," in *proceedings of the IEEE international joint conference on neural networks*, vol. 2, 2002, pp. 1254–1159.

[27] F. R. Kschischang, B. J. Frey, H.-A. Loeliger, *et al.*, "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory*, vol. 47, no. 2, pp. 498–519, 2001.

[28] G. Colavolpe and G. Germi, "On the application of factor graphs and the sum-product algorithm to ISI channels," *IEEE Transactions on Communications*, vol. 53, no. 5, pp. 818–825, 2005.

[29] D. Kipnis and R. Diamant, "A factor-graph clustering approach for detection of underwater acoustic signals," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 5, pp. 702–706, 2018.

[30] F. Dellaert, "Factor graphs and GTSAM: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

[31] A. Cunningham, M. Paluri, and F. Dellaert, "DDF-SAM: Fully distributed SLAM using constrained factor graphs," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 3025–3030.

[32] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 467–475.

[33] T. Yu and J. H. Hansen, "A speech presence microphone array beamformer using model based speech presence probability estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009, pp. 213–216.

[34] H. Ye and D. DeGroat, "Maximum likelihood DOA estimation and asymptotic Cramér-Rao bounds for additive unknown colored noise," *IEEE Trans. Sign. Process.*, vol. 43, no. 4, pp. 938–949, 1995.

[35] X. Li, L. Girin, R. Horaud, S. Gannot, X. Li, L. Girin, R. Horaud, and S. Gannot, "Multiple-speaker localization based on direct-path features and likelihood maximization with spatial sparsity regularization," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 10, pp. 1997–2012, 2017.

[36] H. W. Löllmann, C. Evers, A. Schmidt, H. Mellmann, H. Barfuss, P. A. Naylor, and W. Kellermann, "The LOCATA challenge data corpus for acoustic source localization and tracking," in *IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM)*, Sheffield, UK, July 2018.

[37] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," 1993.

[38] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[39] X. Boyen and D. Koller, "Approximate learning of dynamic models," in *Advances in neural information processing systems*, 1999, pp. 396–402.

[40] K. Murphy and Y. Weiss, "The factored frontier algorithm for approximate inference in dbns," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 378–385.

[41] E. A. Habets, "Room impulse response simulator," `https://github.com/ehabets/RIR-Generator`, International Audio Laboratories, Erlangen.

[42] B. J. Frey and D. J. MacKay, "A revolution: Belief propagation in graphs with cycles," in *Advances in neural information processing systems*, 1998, pp. 479–485.

[43] R. Balan and J. Rosca, "Microphone array speech enhancement by bayesian estimation of spectral amplitude and phase," in *IEEE Sensor Array and Multichannel Signal Processing Workshop Proceedings*, 2002, pp. 209–213.

[44] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 4, pp. 1462–1469, 2006.

[45] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," *IEEE Transactions on information theory*, vol. 51, no. 7, pp. 2282–2312, 2005.

תקציר

בעבודה זו אנו מציגים מספר אלגוריתמים לעקיבה והפרדה של מספר דוברים תוך שימוש במערך מיקרופונים. ראשית אנו מניחים דלילות של הדיבור במרחב הזמן-תדר, כך שבכל רכיב זמן-תדר פעיל דובר אחד לכל היותר. שנית, אנו מניחים מספר סופי של כיוונים אפשריים עבור הדוברים - ״מועמדים״. תוך שימוש בהנחות אלו, אנו ממדלים את הבעיה כבעיית שערוך סטטיסטי שבה אנו מעוניינים לשערך באמצעות המדידות את המשתנים החבויים (hidden data) שהם: 1) כיוון הדוברים (DOA – Direction of Arrival), ו- 2) מסכת זמן-תדר לכל דובר המחליטה עבור אותו דובר באילו רכיבי זמן-תדר הוא היה פעיל.

לאחר מכן אנו מציגים מודל סטטיסטי למדידות בהנחה שהמשתנים החבויים ידועים, ולבסוף אנו מציגים שלושה מודלים סטטיסטיים שונים עבור המשתנים החבויים, ועבור כל אחד מהם עושים שימוש באלגוריתם ייחודי לשערוך סטטיסטי של המשתנים החבויים מתוך המדידות.

בסופו של דבר מוצגים בעבודה זו שלושה אלגוריתמים שונים לפתרון הבעיה, המפורטים להלן.

מודל מיידי (Instantaneous model)

במודל זה אנו מתייחסים למיקום של הדובר הפעיל בכל רכיב זמן-תדר כמשתנה חבוי. מיקום זה מתפלג א-פריורית עם פילוג לא ידוע על פני המיקומים האפשריים. כעת מטרת האלגוריתם היא לשערך פילוג זה, ואנו משתמשים לצורך כך באלגוריתם Expectation-Maximization (EM). אנו מציעים שתי גרסאות לאלגוריתם. בגרסה הראשונה אנו מניחים מצב סטטי ולכן פותרים עבור כל המידע בו-זמנית (Batch-EM), ובגרסה השנייה אנו מניחים דוברים נעים ולכן מציעים פתרון רקורסיבי המחליק את השערוך על פני הזמן (Recursive-EM).

מודל מרקובי (Hidden Markov Model)

גם במודל זה אנו מתייחסים למיקום של הדובר הפעיל בכל רכיב זמן-תדר כמשתנה חבוי. כאשר אנו מתייחסים לכל תדר כאל שרשרת מרקובית, כך שההסתברות שהמיקום של הדובר הפעיל תשתנה עבור אותו תדר לאורך הזמן קטנה מההסתברות שאותו דובר ימשיך להיות פעיל באותו תדר. כיוון שהתדר הפעיל יכול להשתנות עם הזמן אנו מציעים הרחבה של מודל מרקובי מצומד (Coupled Hidden Markov Model) המאפשר מעבר בין תדרים לאורך הזמן בהסתברות מסוימת. לצורך ההסקה סטטיסטית, אנו משתמשים באלגוריתם Forward-Backward, עם הרחבה למקרה של צימוד בין השרשראות המרקוביות.

מודל Factor Graph

במודל זה אנו מציעים שיטה ייחודית לפתרון הבעיה תוך שימוש ב- Factor Graph. ראשית אנו ממדלים כל אחד מהמשתנים החבויים בנפרד, כאשר עבור מיקום הדוברים אנו משתמשים במודל מרקובי ועבור שיוך רכיבי הזמן-תדר לדוברים אנו מציעים להשתמש בפילוג אחיד על פני הדוברים השונים ושווה פילוג בין כל רכיבי הזמן-תדר. לאחר מכן אנו מראים שהפילוג המותנה של המשתנים החבויים בהינתן המדידות ניתן לפירוק למכפלה של תת-פונקציות (פונקציות ״פוטנציאל״). צורת פילוג זו נקראת Factor Graph.

אחת השיטות לשערוך סטטיסטי במודל זה היא Loopy Belief Propagation (LBP). על בסיס שיטה זו פיתחנו אלגוריתם ייחודי להסקה סטטיסטי, הפותר את בעיית ההפרדה ובעיית האיכון בו זמנית, תוך שימוש בכמה משוואות עדכון פשוטות.

תוצאות

עבור כל אחת מהשיטות הראינו תוצאות טובות יותר מהשיטות הקיימות גם עבור הקלטות שנוצרו על ידי סימולציה, וגם עבור הקלטות אמתיות שהקלטנו במעבדה.

עבודה זו נעשתה בהדרכתו של פרופ׳ שרון גנות

מהפקולטה להנדסה של אוניברסיטת בר-אילן

אוניברסיטת בר-אילן

עקיבה והפרדה של דוברים על ידי שימוש בשיטות להסקה סטטיסטית

קובי ויסברג

עבודה זו מוגשת כחלק מהדרישות לשם קבלת תואר מוסמך
בפקולטה להנדסה של אוניברסיטת בר- אילן

רמת גן   תשפ״א