Friedrich-Alexander-Universität Erlangen-Nürnberg

Lehrstuhl für Multimediakommunikation und Signalverarbeitung

Prof. Dr.-Ing. Walter Kellermann

ASC Master's Programme: Major Project

Misalignment Recognition in Acoustic Networks Using a Semi-Supervised Source Estimation Method and Markov Random Fields

Gabriel F Miller

May 2020

Supervisors: Andreas Brendel, and Sharon Gannot

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

_____ _____

Ort, Datum Unterschrift

Contents

Abstract

In this report, we consider the problem of acoustic source localization when an array in a network is potentially compromised. Localization of acoustic sources is done via a semi-supervised approach on multiple manifolds. A Markov Random Field (MRF) model is used to probabilistically asses the disparity in position estimates made by sub-networks of microphone arrays. The latent class posteriors output by the MRF are used to determine if an array is compromised, whereby the latent classes of the MRF models were chosen to represent the possibility of a sub-network position estimate being aligned, misaligned or unreliable with respect to the noise inherent to the estimator. We also assume the latent classes to be fully connected, ensuring the entire relation between sub-networks is considered. This probabilistic approach is advantageous over naive estimations as it outputs a normalized value indicating if there is a misalignment in a given network whereby the value encapsulates conditional information provided by each sub-network. Experimental results show that the performance of the proposed method is consistent in identifying compromised arrays for different levels of disruption in the network.

Chapter 1

Introduction

Sound source localization is a topic that has been covered in great detail for many years and still remains a burgeoning field of study. Some of the different applications that continue to drive the need for more robust and efficient estimators include video conferencing, automatic camera steering, smart home technology and speaker separation [LGTG17a, YCH02, RK89, KC76, DH12, DFH13, DFH15, MPK11, LGTG16, LGTG17b]. While traditional methods of localization can be robust in conditions where noise is not a factor, new methods that utilize learning-based techniques have been developed to produce accurate localization results independent of the noise or reverberation of the given acoustic environment [DH12, DFH13, DFH15, MPK11, LGTG16]. While these techniques are more robust in adverse scenarios, one particular drawback that arises is that learning methods generally depend on the array network dynamics used during a training process. If the dynamic were to change (e.g. the microphone arrays used in training are moved or become compromised in some way), it may be the case that an adjustment needs to be made in how localization takes place in light of the changing dynamics. This of course requires the ability to determine consistently and reliably whether something has changed. This is the problem we consider in this report.

In the past, traditional localization methods have generally been categorized in three different groupings. One such being those based on the maximization of a steered response power (SPR) of a beamformer output whereby the measured signals are filtered

and summed together and the maximum likelihood (ML) criterion is used to determine the output power of a beamformer steered to different locations [Sch86]. There are also high-resolution spectral estimation techniques, such as the multiple signal classification (MUSIC) method [YCH02], or the estimation of signal parameters via rotational invariance (ESPRIT) [RK89] algorithm. Both are based on the spectral analysis of the correlation matrix of measured signals. A third class of methods are a dual-stage approaches relying on time difference of arrival (TDOA) estimates. In the first stage, the TDOAs of different pairs of microphones are estimated and collected with each reading of the TDOA said to correspond with the single-sided hyperbolic hyperplanes (in 3D) representing possible positions. The intersection of these hyperplanes corresponding to each sensor pair yields the estimated position. One classic TDOA estimation method of this type is the generalized cross-correlation (GCC) algorithm [KC76].

What most of these methods have in common, is that they rely on simplified physical models that are dependent on assumptions regarding the propagation model, e.g. far field and free field, and the statistics associated with signals that are trying to be localized [LGTG16]. Accurate physical models though, have a level of complexity that make robust estimates very difficult to acquire. For example, in an environment with adverse audio conditions, such as high reverberation levels, or a room with a non-uniform structure and composed of many different materials with varying reflective qualities, getting a robust estimate becomes more difficult.

More recently, there has been a new found interest in estimating an acoustic source using learning-based methods, whereby position estimates are obtained directly from data. These methods can generally be classified as either supervised or unsupervised, with supervised implying that in a training phase, source positions as well as the signals emitted and received at each microphone array are known, whereas with unsupervised methods, only measurements are available. Such methods have been proposed for both the binaural and multiple microphone array cases [DH12, DFH13, DFH15].

In both array classes, different features have been explored. In [MPK11], the authors used a Gaussian Mixture Model (GMM)-based approach to learn the azimuth-

dependent distribution of the binaural feature space. In [LGTG16], a power spectral density (PSD)-based feature vector with a so-called semi-supervised learning approach is discussed. The semi-supervised method is said to be preferred for the task of source localization (over a supervised approach) as the amount of data available is limited to the acoustic environment being considered (unlike in other recognition scenarios where a universal model can be built from a general database of examples). In other words, the training needs to fit the specific acoustic environment in which measurements are obtained, and thus, we cannot create a general database that corresponds to all possible acoustic scenarios. Instead, the training set needs to be generated individually for each acoustic environment [LGTG16]. To obtain labelled data, one needs to generate recordings in a controlled manner and calibrate each of them precisely, making the process of generating a large amount of labelled data cumbersome and impractical. Thus, in order to have a larger database to work with for any particular scenario, unlabelled data is used, as it is less cumbersome to acquire (i.e. it can be collected whenever someone is speaking), and still a useful feature for learning.

In [LGTG17b], a multiple manifold-based approach with a semi-supervised inference method for acoustic localization was developed. In this approach, each manifold corresponds to an array, and both labelled and unlabelled acoustic sources are jointly used in training to localize a dense grid of relative transfer functions (RTFs). This is done by applying a semi-supervised Gaussian process (SSGP) [SCK07,LGTG17b]. The method essentially looks to estimate a target function (i.e. a regression function that maps an RTF function to a sound source position) using a Bayesian inference framework in which the likelihood function measures the correspondence of the target function to labelled sources, while the prior probability reflects the prior belief regarding the distribution of the target function. The prior relies on a manifold model, meaning it relies on the geometric structure of RTFs corresponding to labelled and unlabelled samples whereby the labelled sources act as an anchor for future estimates of a given test position [LGTG17b], and unlabelled samples act as an alternate measure of complexity of the manifold, (i.e. they measure smoothness with respect to different data manifolds

or clusters) [SCK07]. A manifold kernel function is used to capture the complexity of the manifold in a training phase and gain information on the characteristics of a given acoustic environment. The retained information is used to infer the position of any source during a test phase.

We consider the semi-supervised localization approach with a static sound source where any given microphone array in an array network is compromised. In particular, we assume array movement to be the cause of the changing dynamics of the network. We know that the estimation of a source position with the semi-supervised method depends on the dynamics of the network of nodes used in training to remain consistent, i.e. if a node moves, the RTF samples that describe the room dynamics from the perspective of a given node may not be relevant for estimating a new position estimate and can corrupt the overall network estimate (more details provided regarding the method in chapter 2). In order to determine if movement occurs, we consider a technique recently introduced in the field of robotics for recognizing sensor misalignment [AMYHM19]. Here, the authors utilize Markov Random Fields (MRFs) with fully connected latent variables to measure the probability of a sensor network being misaligned based on individual sensor readings and a ground truth mapping of a given room. Recognition of misalignment is needed in their case to determine whether differences in measurements over time should be attributed to potential changing dynamics within a room or due to inherent noise. For our problem, rather than taking each sensor signal independently, we look at leave-one-node-out (LONO) sub-network positional estimates (with each sub-network containing all but one array), with estimates obtained via the semi-supervised method. We then use the error in position estimates of a sound source from LONO sub-networks as input to the MRF model. In the end, our model outputs posterior probabilities per sub-network for belonging to one of the following latent states: aligned, misaligned or uncertain (the uncertainty class accounts for the inherent error of the semi-supervised estimator). These posteriors can then be used to indicate both the probability of some dynamic changing in the network, and also allows for inference of which array in particular may be compromised (i.e. a potentially compromised array is one whereby

the sub-network that does not include said array has latent probabilities that indicate alignment).

In order to motivate the use of a probabilistic approach, we look at Fig. 1.1. In the top panels we see the mean and standard deviation in error of positional estimates of a static sound source via the semi-supervised method. These experiments were done for increasing shifts of a randomly selected array (shifted relative from where the array was during training) and varying T60s. The error is compared to the average error with the same T60 with no movement. Note that the average and standard deviation in error from the semi-supervised method after an array has moved can increase quite significantly (e.g. the error is nearly twice as high for a T60 of 0.2 seconds (s) and a shift of 1.05 meters). This of course indicates something needs to be done to update the estimator in light of the movement, which in turn requires some consistent indicator of when movement occurs. In the bottom panels, we see the average and standard deviation of the probability of movement output by our proposed method for the same set of array shifts. Note that with the probabilistic approach, there is a clear correlation with the increase in shift and probability of misalignment. This gives more flexibility in determining how sensitive a model should be to movement (sensitivity controlled by varying a given probability threshold).

The remainder of this report is structured as follows. Chapter 2 provides detail regarding the semi-supervised estimation method we consider. In chapter 3, more background and details are given regarding the MRF-based detection method. In chapter 4, we quantify the ability of our approach and compare it to some naive methods, and in chapter 5 we conclude with a summary of our findings.

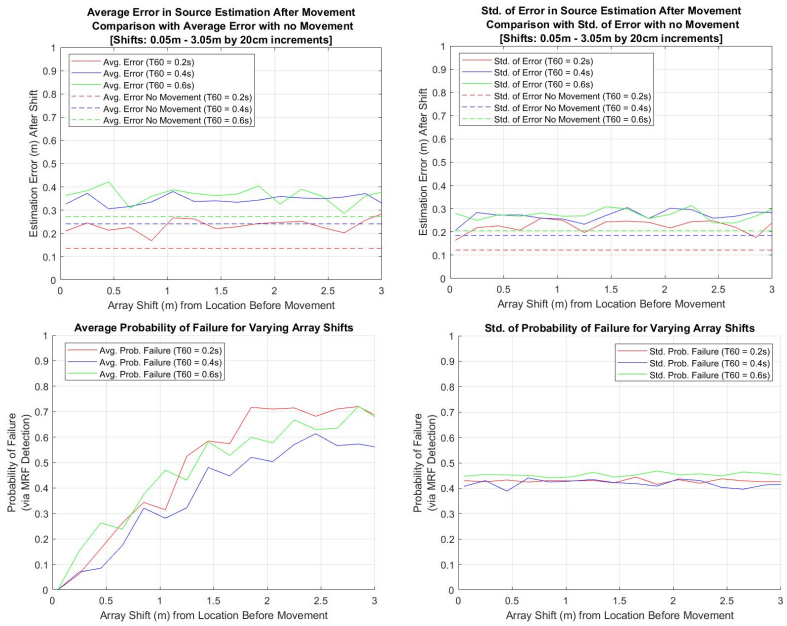


Figure 1.1: Average and standard deviation of error in position estimation per the semi-supervised localization estimator for increasing distance in array shifts (top left, top right panels respectively) with the average error of the estimator with no array movement included as reference. In the bottom left and right panel we see the average and standard deviation respectively of the probability of failure output by the MRF detection algorithm.

Chapter 2

Semi-Supervised Acoustic Source Localization

As noted, in this report we consider the same approach to estimating the position of an acoustic source using multiple microphone arrays as that done in [LGTG17b]. The authors utilize a feature vector based on relative transfer function (RTF) estimates, and a so-called manifold-based prior, meaning it relies on the geometric structure of RTF samples implied by labelled and unlabelled samples [LGTG16]. The complexities associated with both types of data are captured via a manifold kernel, which is useful for identifying structural similarities in non-linear classification problems. This is done based on an analysis of a low-dimensional intrinsic geometry describing the acoustic scene, driven by measurements (RTF samples in our case) [SCK07]. The acquired information gained during a training phase regarding the acoustic characteristics of a given environment are used to infer the position of an unknown source during a test phase.

2.1 Background

We look to estimate a single source position $\mathbf{q} = [q_x, q_y, q_z]^T$ using an ad-hoc network of microphone arrays where each array consists of 2 microphones spaced 5 cm apart

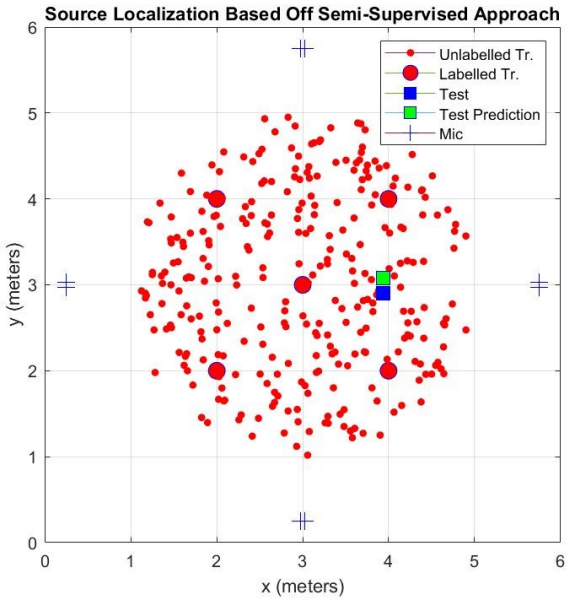


Figure 2.1: Overview of the simulated room used for estimating a test audio source based on a semi-supervised approach using labelled and unlabelled training (tr.) positions.

(see Fig. 2.1). The source produces an unknown speech signal $s(t)$, which is recorded by all microphones. The signal received by the ith microphone of he mth pair is given by:

$$y_i^m(t) = a_i^m(t, \boldsymbol{q}) * s(t) + u_i^m(t). \tag{2.1}$$

Here, m references the microphone array and i specifies one of the two microphones in the mth array. Additionally, $a_i^m(t, \boldsymbol{q})$ is the acoustic impulse response (AIR) relating the source at position $\mathbf{q}$ and the ith microphone, and $u_i^m(t)$ is an additive noise signal which corrupts the corresponding measured signal. Linear convolution is denoted by *. We can see that the information required for localization is embedded in the AIR, and is independent of the source signal. Thus the goal presented in the semi-supervised approach to localization is to extract a feature vector $\mathbf{h}^m$ that depends solely on the two AIRs of the corresponding node during a training phase, to later estimate the position of a test source based off it's corresponding feature vector.

2.2 SSGP Estimation

2.2.1 RTFs

As noted, the feature vector used is the so-called RTF. RTFs are typically represented in a high dimensional space, with a large number of coefficients to allow for the full description of the acoustic paths between two microphones, which represents a complex reflection pattern [LGTG17b,LTG13]. In reality though, the RTFs are controlled by a small set of parameters, such as room dimensions, reverberation time, location of the source and microphone arrays, etc. This of course naturally leads to the assumption that a large part of the information that RTFs hold are confined to a low dimensional manifold. Moreover, because training is done for a particular enclosure, in essence the room dynamics (e.g. dimensions, reverberation) are cared for. Thus, the only outstanding variable subject to change is the position of a given acoustic source that we look to localize. The mapping function used to relate the high dimensional RTF samples and the source positions, is modelled as a Gaussian process with a covariance function that is based on a Gaussian kernel function.

In particular, consider the function $f_a^m : M_m \to \mathbb{R}$ for $a \in \{x,y,z\}$ whereby f_a^m maps an RTF sample, $\mathbf{h}^m$, associated with node, m, to the corresponding coordinates of the source position. We denote the position evaluated by f_a^m for the RTF sample $\mathbf{h}^m$ as p^m (i.e. $p^m \equiv f_a^m(\mathbf{h}^m)$ where we forego use of a as the same estimation technique is used for each coordinate. It is assumed that p^m follows a Gaussian process (GP) which is used in practice as a method of analyzing data in the supervised learning context [ER04,LGTG17b,LTG13]. There are many benefits to GPs, e.g., while general parametric models can lack in expressive power or intelligibility, GPs are thought to excel [ER04]. In addition, GPs are nice in that they essentially fit data and complexity terms automatically. Weights do not require to be trained, and cross validation is also not necessary. By modelling the mapping function from the RTF samples to the source positions as a GP, we have an easy and intuitive way of updating a covariance matrix that describes the relation between the signals observed at each node as new

information is presented [ER04].

In order to express the complex relation tying together all the information obtained at each node, the semi-supervised method utilizes a kernel-based covariance matrix, whereby each element represents a pairwise affinity between two RTF samples. In particular, we express the relationship between two processes in terms of the manifold-based kernel:

$$\begin{aligned}\tilde{k}_m(\mathbf{h}_i^m, \mathbf{h}_j^m) &\equiv \mathrm{cov}(p_{l_i}^m, p_{l_j}^m) \\ &\coloneqq \sum_{i=1}^{n_D} k_m(\mathbf{h}_{l_i}^m, \mathbf{h}_i^m) k_m(\mathbf{h}_{l_j}^m, \mathbf{h}_i^m), \quad \forall l_i, l_j\end{aligned} \tag{2.2}$$

where we note that l_i and l_j denote ascription to labelled source positions, n_D denotes the total number of RTF samples used in training (including both labelled and unlabelled samples), and k_m ($\mathbf{h}_i^m$, $\mathbf{h}_j^m$) is a standard pairwise kernel function k_m: $M_m \times M_m \to \mathbb{R}$. In particular, we put to use the Gaussian kernel:

$$k_m(\mathbf{h}_i^m, \mathbf{h}_j^m) \coloneqq \exp\left(-\frac{\left\|\mathbf{h}_i^m - \mathbf{h}_j^m\right\|^2}{\varepsilon_m}\right). \tag{2.3}$$

As noted, the manifold kernel is useful for identifying structural similarities in non-linear classification problems. To further understand this, we consider some of the theory behind the kernel as was introduced in [SCK07] which introduces the kernel in the context of semi-supervised GP (SSGP) classification. The paper details the SSGP as an analysis scheme that considers a low-dimensional intrinsic geometry driven by measurements. The analysis relies on a data set with many features that can be expressed by a lower dimensional representation, and acts to preserve the geometric relationship when mapping from the original space to the lower dimensional one. The hope is that this new representation will capture the main structures of the data in fewer dimensions, thereby achieving dimensionality reduction [TCGC13].

One of the benefits of the semi-supervised approach is that unlabelled data, which is

typically much easier to acquire, is used effectively. It is said unlabelled data may suggest alternate measures of complexity, such as smoothness with respect to data manifolds or clusters, that can be used to re-structure the Hilbert space, $\mathcal{H}$. Here, $\mathcal{H}$ is a deterministic Reproducing Kernel Hilbert Space (RKHS) and is closely related to a Hilbert space of random variables spanned by the GP via a classical isometry [SCK07]. Note that the two are related via the kernel covariance function, whereby the function is the covariance of the GP and also the kernel function of $\mathcal{H}$. In essence, the SSGP approach refines the norm describing the difference in observations at each node, by combing the original ambient smoothness (from the labelled data) with an intrinsic smoothness measure (from the unlabelled data) defined in terms of the kernel covariance matrix [SCK07, CPRD16].

We now look at how to practically measure smoothness with respect to the manifold and infer similarities between new test samples and clusters defined in the lower dimensional space. This is realized via the kernel-based covariance matrix, which utilizes information gained from both labelled and unlabelled training points. We first formally define the kernel covariance matrix, $\tilde{\mathbf{\Sigma}}_L \subseteq \mathbb{R}^{n_L \times n_L}$, that consists of elements based off kernel relations we have already introduced, with n_L denoting the labelled points used for training. In addition, note that the covariance takes into account, and essentially blends the viewpoint of each microphone array. This is done by averaging the product of all observed RTF samples for all M^2 possible combinations of nodes [LGTG17b]. In particular we say

$$\begin{aligned}\left(\tilde{\Sigma}_L\right)_{l_i,l_j} = \tilde{k}\left(\mathbf{h}_{l_i}, \mathbf{h}_{l_j}\right) &\equiv \operatorname{cov}(p_{l_i}, p_{l_j}) \\ &= \frac{1}{M^2}\sum_{d=1}^{n_D}\sum_{q,w=1}^{M} k_q\left(\mathbf{h}_{l_i}^q, \mathbf{h}_d^q\right) k_w\left(\mathbf{h}_{l_j}^w, \mathbf{h}_d^w\right)\end{aligned} \tag{2.4}$$

where we denote d to indicate indexing over all RTF samples (labelled and unlabelled), q, w refer to microphone arrays, and p_{l_i}, p_{l_j} refer to labelled points. It should now be clear how the labelled points and their associated transfer functions play as a sort of anchor for realizing a given GP, while the unlabelled points still play a key role in

re-structuring the matrix based off additional information [LTG13, LGTG17b].

Interestingly, with some simple derivations, one can find an alternative definition for the covariance matrix, $\tilde{\mathbf{\Sigma}}_L$, namely:

$$\tilde{\mathbf{\Sigma}}_L = \frac{1}{M^2} \sum_{q,w=1}^{M} \mathbf{K}_L^q \mathbf{K}_L^w \tag{2.5}$$

where each element in $\mathbf{K}_L^m$ for a given microphone array is calculated via (Eq. 2.3).

This formulation allows us to see more clearly how the information obtained by different nodes is aggregated. Essentially, by multiplying the kernels of each node pairing as indicated in Eq. 2.5, we are essentially averaging out incoherent node-specific variables and are left with only the common variable, which is the position of the source. That is to say, when measuring the correlation between two nodes the common source of variability is emphasized, i.e. the source position, and we suppress artifacts and interference, which are node specific effects [LGTG17b].

With the derivation of the covariance matrix containing the weights relating the viewpoints of each array, it is now possible to provide a robust estimate of a source test position given the RTF sample associated with the test source, $\mathbf{h}_t$. This is done, by updating our weights in a Bayesian manner based off the information provided by the new sample [LGTG17b]. First, we denote positional estimates as follows:

$$\bar{p_i} = p_i + \eta_i; i = 1, \ldots, n_L \tag{2.6}$$

with $\eta_i \backsim \mathcal{N}\left(0, \sigma^2\right)$ being i.i.d. Gaussian noises, independent of p_i. Moreover, with p_i and η_i independent, and jointly Gaussian, this implies $\bar{p}_i$ is also jointly Gaussian. Thus when considering some new test RTF sample, $\mathbf{h}_t$ that is the product of some unknown source at an unknown location we can solve for the position estimate based on the posterior probability:

$$\mathbb{P}\left(p_t \equiv f(\mathbf{h}_t) \mid \boldsymbol{p}_L, \mathbf{H}_L\right) \tag{2.7}$$

with $\boldsymbol{p}_L = \left[\bar{p}_1, \ldots, \bar{p}_{n_L}\right]^T$, $\hat{p}_t$ denotes the unknown position of a test source, and $\mathbf{H}_L = \{\mathbf{h}_i\}_{i=1}^{n_L}$.

We can thus construct an estimate for the unknown position of a test source, $\hat{p}_t$, based on the maximum a posteriori probability (MAP) estimator, which is equivalent to the minimum mean squared error (MMSE) estimator in the Gaussian case [LGTG17b]. To do so, first note that the labelled training positions, $\boldsymbol{p}_L$ are jointly Gaussian with:

$$\begin{bmatrix} \boldsymbol{p}_L \\ p_t \end{bmatrix} | \mathbf{H}_L \backsim \mathcal{N}\left(\mathbf{0}_{n_L}, \begin{bmatrix} \tilde{\boldsymbol{\Sigma}}_L + \sigma^2 \mathbf{I}_{n_L} & \tilde{\Sigma}_{Lt} \\ \tilde{\Sigma}_{Lt}^T & \tilde{\Sigma}_t \end{bmatrix} \right). \tag{2.8}$$

We've already introduced $\tilde{\boldsymbol{\Sigma}}_L$ as an $n_L \times n_L$ covariance matrix defined over function values at labelled points. Additionally, we define $\tilde{\Sigma}_{Lt}$ to be an $n_L \times 1$ covariance vector between the function values at $\mathbf{H}_L$ and some function value at a test point p_t, $\tilde{\Sigma}_t$ is the variance of p_t, σ^2 is the variance associated with the Gaussian noise, and $\mathbf{I}_{n_L}$, $\mathbf{0}_{n_L}$ are the $n_L \times n_L$ identity matrix and null matrix respectively. With this in mind, we can now define the positional optimal estimate of some test position,

$$\hat{p}_t = \mu_{cond} = \tilde{\Sigma}_{Lt} \tilde{p_L}. \tag{2.9}$$

Here, μ_{cond} is the mean of the multivariate Gaussian distribution,
$\mathbb{P}\left(p_t \mid \boldsymbol{p}_L, \mathbf{H}_L\right)$, and is calculated by multiplying the test covariance vector by a vector of weights, $\tilde{\boldsymbol{p}}_L$, which are independent of a given test sample and are obtained via the inverse covariance matrix with some added noise:

$$\tilde{\boldsymbol{p}}_L = \left(\tilde{\boldsymbol{\Sigma}}_L + \sigma^2 \mathbf{I}_{n_L}\right)^{-1} \boldsymbol{p}_L. \tag{2.10}$$

2.3 SSGP Procedural Overview and Results

Below we give a general overview of the localization procedure [LGTG17b]:

Training Phase

- *RTF estimation:* We first estimate RTF samples for both labelled and unlabelled sources. Samples are recorded for each microphone array.
- *Covariance Estimation:* The covariance matrix, which associates the different samples recorded at each array, is calculated. It is with the inverse covariance (precision) matrix that we estimate the position of a test source.

Test Phase

- *RTF estimation:* We first estimate the RTF associated with a given test source.
- *Covariance Estimation:* A covariance vector is calculated relating the test RTF measured at each array with those used for training.
- *Adaptation:* Here we adapt the the precision matrix to account for the new RTF (test) sample.
- *Position Estimation:* The updated precision matrix is used to estimate the position of the test source.

For our considerations, we simulate a room with dimensions $6 \times 6 \times 3$ m (Fig. 2.1) with the left lower corner of the room as the origin of the Cartesian coordinate system. For specific details regarding the setup see section 4.1.

The semi-supervised algorithm was tested under varying room dynamics (i.e. T-60s ranging from 0.15 s - 0.65 s by increments of 0.05 s). The mean, μ_{ssgp} and standard deviation, σ_{ssgp}, of the SSGP estimator error were recorded (Table 2.1). This was done by drawing at random speech signals from a database of English speakers [Pov18], randomizing the position of the source and comparing the positional estimates to the ground truth.

Table 2.1: Mean and standard deviation of error for the semi-supervised source localization algorithm for different T60s (mean and standard deviation reported is in meters, T60 in seconds).

T60s	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	0.6	0.65
μ_{ssgp}	0.09	0.13	0.15	0.19	0.21	0.24	0.25	0.26	0.27	0.27	0.26
σ_{ssgp}	0.09	0.12	0.13	0.15	0.17	0.18	0.2	0.19	0.2	0.21	0.2

Chapter 3

Network Misalignment Detection

In order to determine if a network of nodes is aligned (i.e. all nodes are in the same position they were during training of the SSGP estimator), there are several components that need to be considered. Firstly, there must be some way of knowing when a given array becomes compromised, as well as knowing which array it was. In this paper, we consider these problems via a technique recently introduced in the field of robotics for recognizing sensor misalignment using Markov Random Fields (MRFs) with fully connected latent variables (FCLVs) [AMYHM19]. In their work, the authors consider the problem of discerning whether individual sensor readings for sensors in a given network are aligned to localize a given object. In particular, a set of sensors is used to compare physical readings of an open space with a map of said space that is known beforehand. The method implemented takes as input the localization error of a given sensor reading and that of the true map, and estimates the posterior class probabilities of each sensor measurement via an MRF with FCLVs. The FCLV property allows for consideration of the entire network relationship to indicate whether sensor measurements are aligned, misaligned or obtained from unknown obstacles (not included in the original mapping). The posterior probabilities output from the MRFs with FCLVs are used to indicate when there is a misalignment among the sensors, and to detect which particular sensor led to localization failure.

3.1 Markov Random Fields

MRFs provide a convenient and consistent way of modeling context dependent entities, in particular spatially correlated features, and are better suited than directed networks in expressing soft constraints between random variables [Bis06]. Moreover, with respect to computational costs MRFs can be implemented in a local and massively parallel manner [Li95]. In practice, MRFs assume some observed quantities y_i from which we look to infer hidden variables that dictate the behavior of the observations, and which we denote as x_i. Note that the index i can be thought of as representing a node position, or the position of a small patch of nodes, where each neighborhood, i, is potentially governed by a different distribution.

We now define the probabilistic relation between hidden states and our observations as $\phi_i(y_i, x_i)$, which is often referred to as the evidence of y_i. It's worth noting that for some of the more common use cases of MRFs, e.g. computer vision and robotics, the function usually denotes a disparity between an observation and a given set of labels. For example, in image restoration the observations are typically a pixel intensity while the labels are a discrete and often smaller set of possible intensities [BVZ01], and in [AMYHM19], the labels used were localization estimates from a ground truth mapping with the observations being actual sensor estimates. As noted, the observations for our problem are positional estimates measured by all possible LONO sub-networks of microphone arrays with the latent labels dictating whether the state of a given sub-network is aligned, misaligned or uncertain. We also define the potential function which models dependencies in local groupings, and imposes structure amongst the hidden variables. We denote the function as $\psi_{i,j}\left(x_i, x_j\right)$ which defines the potential of a given set of local nodes being governed by a certain distribution [Bis06]. In our case, this would be the observations from each sub-network of nodes belonging to a given latent state, each with its own associated distribution based on the frequency of occurrence of a given error size as will be detailed below. We define the joint probability as follows:

$$\mathbb{P}\left(\mathbf{x},\mathbf{y}\right)=\frac{1}{Z}\prod_{(i,j)}\psi_{i,j}\left(x_i,\ x_j\right)\prod_{i}\phi_i\left(x_i,y_i\right) \tag{3.1}$$

where Z is a normalization constant and the product over (i,j) is over nearest neighbors.

We've now made note on how the factorization of the potential functions and the structure of conditional independence inherent to a given network are useful for defining a set of probability distributions within a given network, i.e. we compute the marginals for neighborhoods of nodes, i, in order to infer something about the underlying relationship governing our network. We now look at how this is realized in practice. This realization was proven by John Hammersley and Peter Clifford [HC71]. The theorem they proved essentially says if a network defined by a given probability distribution that satisfies one of the Markov properties (see Appendix A), and that are all strictly positive, then can be factorized over the set of maximal cliques (cliques being neighborhoods of nodes in a graph such that there exists a link between all pairs of nodes in the subset, and maximal implies inclusion of any other node in the network to the clique would lead to the neighborhood no longer being a) [LM01, Bis06]. With this in mind, we can now define the joint probability defined in Eq. 3.1 as the product of potential functions over the set of maximal cliques, $\mathcal{C}$, of the graph [Bis06]:

$$\mathbb{P}\left(\mathbf{x},\mathbf{y}\right)=\frac{1}{Z}\prod_{\mathcal{C}}\psi_C\left(x_C\right). \tag{3.2}$$

In general, potential functions are defined in terms of Gibbs (thermal) states [KB19] with $\psi_C\left(x_C\right)=\exp\{-\mathcal{E}\left(x_C\right)\}$. Here, $\mathcal{E}$ is an energy function and acts as an indicator of the likelihood of corresponding relationships within a given clique, with a higher energy configuration having lower probability and vice-versa [TBF05]. Thus, to maximize the posterior, we look for the set of maximal cliques with the corresponding minimum total energy [Bis06].

3.1.1 Inference Methods for Graphical Models

Inference on graphical models is the task of estimating the posterior distributions of neighborhoods of latent variables based off of observations [Bis06]. In practice, several algorithms have been developed to maximize the posterior probability (i.e. minimize the total energy) with the choice of algorithm dependent on the assumed graphical model.

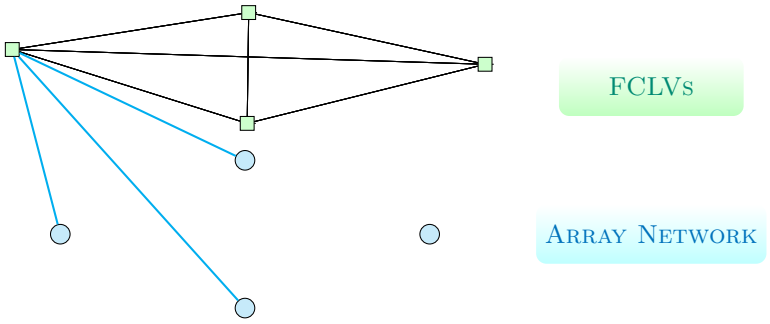


Figure 3.1: Graphical Model with FCLVs. For our considerations, latent states imply whether a sub-network of arrays are aligned, misaligned or uncertain with respect to the possible source localizer error.

The inference model we utilize assumes a network that is fully connected and undirected (Fig. 3.1) with M observations (localization difference output between M LONO sub-networks with M nodes) and 3 possible latent states (aligned, misaligned or uncertain). As mentioned, the model takes as input the error in localization estimation of a static sound source as recorded by each LONO sub-network and outputs a set of posterior probabilities indicating if a given sub-network is aligned, misaligned or there is some uncertainty regarding the observations due to the variance inherent to the SSGP estimator. Note that when a sub-network is probabilistically aligned and if all other sub-networks are probabilistically misaligned, one could infer that there is some disturbance in the network likely due to the array not included in the sub-network that was said to be aligned. An example of the detection algorithm can be seen in (Fig. 3.2) with the source position estimate highlighted before and after movement of an array, and the latent class posteriors after movement indicated in brackets.

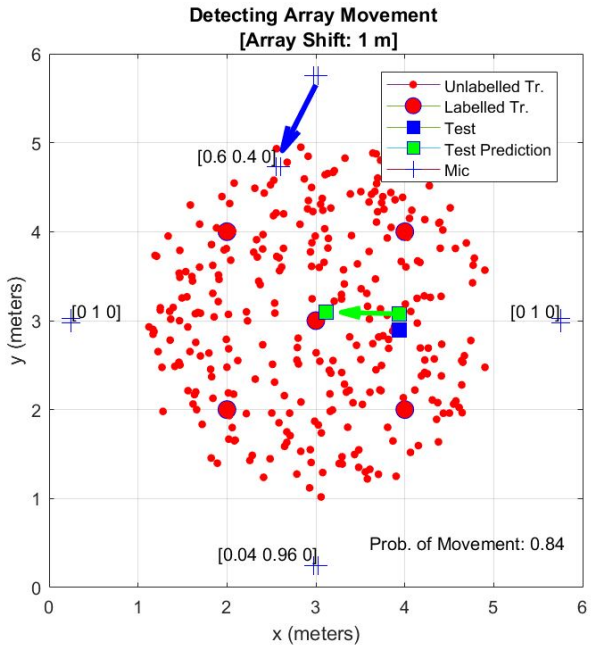


Figure 3.2: Room setup with example of proposed method. Values in brackets next to array nodes indicate probability of a sub-network without the referenced array being aligned, misaligned, or unreliable. Higher probabilities of alignment indicate array left out is likely compromised. Green arrow indicates how the prediction of an acoustic source changes based on the movement and rotation of a random array (blue arrow).

3.2 Latent Class Posterior Probability Estimation

3.2.1 Parameters of the MRF

We begin discussion on how the posterior probabilities are estimated via the MRF algorithm by making note on the potential function, ψ_c (as seen in 3.2). In general, the choice of potential functions for undirected graphs are not restricted to functions that have a specific probabilistic interpretation (in contrast to directed graphs in which each factor represents the conditional distribution of the corresponding variable, conditioned on the state of its parents and must be in the form of a probability). This affords greater flexibility in choosing the potential function for different problems [Bis06]. Generally, the function is chosen in order to reflect configurations of local variables which are preferred to others, specific to the problem at hand. We optimize the function via the iterative proportional fitting procedure (IPFP) [Bis06, Mur12, FM81], a commonly used algorithm for ML estimation in log-linear models. The simplicity of the algorithm and applicability to the analysis of cross-classified categorical data or contingency tables make it a useful tool for optimization of a potential function [FM81]. For specific details on how the parameter was optimized, see 4.2.2.

We now consider notation regarding the prior distributions associated with each latent class. Note that each latent variable is a binary value corresponding to whether a sub-network m belongs to a given latent state i ($z_{m,i} \in \{0, 1\}$). Formally, the distributions of the priors are defined as follows:

$$\begin{aligned}
\mathbb{P}\left(e_m \mid z_{m,1} = 1, \theta_1\right) &= 2\mathcal{N}\left(e_m; 0, \sigma^2_{align}\right), \\
\mathbb{P}\left(e_m \mid z_{m,2} = 1, \theta_2\right) &= \frac{\lambda \exp\{-\lambda e_m\}}{1 - \exp\{-\lambda e_{max}\}}, \\
\mathbb{P}\left(e_m \mid z_{m,3} = 1, \theta_3\right) &= \text{unif}\left(0, e_{max}\right),
\end{aligned} \tag{3.3}$$

where e_{max} is the maximum localization difference, θ_l is the hyper-parameter of the l-th likelihood distribution (i.e. $\theta_1 = \sigma^2_{align}$, $\theta_2 = \lambda$ and $\theta_3 = \emptyset$), and e_m is the error in source estimation by a given LONO sub-network, m:

$$e_m = \left\| \hat{p}_{m,t} - \hat{p}_{m,t+1} \right\|^2 . \tag{3.4}$$

The choice of distribution corresponding to each latent state was derived from empirical data (for more details on how the prior distributions were determined and calculated see section 4.2.1).

3.2.2 Posterior Estimation via Marginalization

The latent probabilities are defined as follows:

$$\mathbb{P}\left(\mathbf{Z}\right) = \left[\mathbb{P}\left(\mathbf{z}_1\right), \mathbb{P}\left(\mathbf{z}_2\right), \ldots, \mathbb{P}\left(\mathbf{z}_M\right)\right] \tag{3.5}$$

with $\mathbb{P}\left(\mathbf{z}_m\right)$ defined as:

$$\mathbb{P}\left(\mathbf{z}_m\right) = \left[\mathbb{P}\left(\mathbf{z}_{m,1} = 1\right), \mathbb{P}\left(\mathbf{z}_{m,2} = 1\right), \mathbb{P}\left(\mathbf{z}_{m,3} = 1\right)\right]. \tag{3.6}$$

In order to define the conditional posterior, $\mathbb{P}\left(\mathbf{Z} \mid \mathbf{e}\right)$, with $\mathbf{e}$ denoting the set of sub-network localization differences, we recall that in theory the posterior can be solved directly via marginalization of the priors. Techniques that compute the conditional probability directly are exact inference methods. However, as was the case in [AMYHM19], we instead use an approximate inference method to avoid the potential computational complexity required to find $\mathbb{P}\left(\mathbf{Z} \mid \mathbf{e}\right)$ (with 3^M possible configurations of probabilities). Instead we consider the marginal posteriors for each sub-network individually:

$$\mathbb{P}\left(\mathbf{z}_m \mid \mathbf{e}\right) = \sum_{\mathbf{z}_1} \cdots \sum_{\mathbf{z}_{m-1}} \sum_{\mathbf{z}_{m+1}} \cdots \sum_{\mathbf{z}_M} \mathbb{P}\left(\mathbf{Z} \mid \mathbf{e}\right). \tag{3.7}$$

3.2.3 Message Passing Scheme

In order to form the optimal set of cliques in our network via marginalization, we utilize a message passing scheme specific to FCLV networks [Bis06]. The procedure

ensures each set of posterior probabilities corresponding to a given sub-network is fully informed by the configuration of probabilities assigned to every other sub-network, which in turn are informed by prior distributions that describe each latent class. The initialization process for the posteriors corresponding to each sub-network is defined as follows:

$$\begin{aligned}\mathbb{P}\left(\mathbf{z}_m \mid \mathbf{e}\right) = \frac{1}{Z}\mathbf{l}_m &\odot \boldsymbol{\mu}'_{1\to m}\left(\mathbf{z}_m\right) \\ &\vdots \\ &\odot \boldsymbol{\mu}'_{m-1\to m}\left(\mathbf{z}_m\right) \\ &\odot \boldsymbol{\mu}'_{m+1\to m}\left(\mathbf{z}_m\right) \\ &\vdots \\ &\odot \boldsymbol{\mu}'_{M-1\to m}\left(\mathbf{z}_m\right)\end{aligned} \tag{3.8}$$

where Z is a normalizing factor ensuring our posteriors are indeed probabilities, $\odot$ is the Hadamard product, $\mathbf{l}_k$ is a likelihood vector, with:

$$\mathbf{l}_m = \left[\mathbb{P}\left(e_m \mid z_{m,1}\right), \mathbb{P}\left(e_m \mid z_{m,2}\right), \mathbb{P}\left(e_m \mid z_{m,3}\right)\right] \tag{3.9}$$

where we've we've shortened our notation from $\mathbb{P}\left(e_m \mid z_{m,l} = 1, \theta_1\right)$ to $\mathbb{P}\left(e_m \mid z_{m,l}\right)$, and finally we say $\boldsymbol{\mu}'_{i\to j}\left(\mathbf{z}_j\right)$ is the message sent from the ith to jth sub-network and is defined as follows:

$$\boldsymbol{\mu}'_{i\to j}\left(\mathbf{z}_j\right) = \psi_c\left(\mathbf{z}_i, \mathbf{z}_j\right) \odot \mathbf{l}_i \tag{3.10}$$

with the likelihood vector elements of $\mathbf{l}_i$ sampled via the Monte Carlo Method [Bis06]. After each node receives the messages from all other nodes, nodes then send messages via the following passing strategy:

$$\begin{aligned}\boldsymbol{\mu}_{m-1\to m}\left(\mathbf{z}_m\right) =& \psi_c\left(\mathbf{z}_{m-1}, \mathbf{z}_m\right)\mathbf{l}_{m-1} \\ & \odot \psi_c\left(\mathbf{z}_{m-2}, \mathbf{z}_{m-1}\right)\mathbf{l}_{m-2} \\ & \vdots \\ & \odot \psi_c\left(\mathbf{z}_{m+2}, \mathbf{z}_{m+1}\right)\mathbf{l}_{m+1}\end{aligned} \tag{3.11}$$

with messages passed continuously until convergence (i.e. the posteriors in consecutive iterations are less than or equal to some threshold).

3.2.4 Localization Failure Detection

With the posteriors for each sub-network, we obtain the probability of misalignment in the overall network based on the average posterior probabilities of misalignment for sub-networks that are probabilistically misaligned:

$$p_{misalign} = \frac{1}{M}\sum_{m=1}^{M}\mathbb{P}\left(\mathbf{z}_{m,2} \mid \mathbf{e}\right). \tag{3.12}$$

We say that movement in the network has occurred when $p_{misalign} \geq p_{thresh}$ whereby p_{thresh} is some probability threshold determined based off the tolerance for movement. Note that in practice the tolerance will depend on the room dynamics and noise at each microphone node (i.e. less tolerance would be preferred in an acoustic environment where localization is more prone to error for smaller movements).

Chapter 4

Experimental Results

4.1 Setup

4.1.1 Room Description

A room with dimensions 6 x 6 x 3 m was simulated (Fig. 3.2) with the left lower corner of the room as the origin of the Cartesian coordinate system. Our setup included 4 arrays spaced roughly 4.47 m apart, each comprised of two microphones spaced 5 cm apart. Our region of interest (RoI), i.e. the region in which both labelled and unlabelled signals originate, sat in the middle of our arrays within a 2 m radius of the center of the room. In total we simulated five labelled sources with four sources uniformly spaced in a square with a side length of 2 meters (m), and a fifth source located in the center of the square (1.41 m away from all other sources). We also used 300 unlabelled sources to generate RTFs whereby each unlabelled point was randomly generated within the aforementioned radius.

4.1.2 Details on Signal, RIR Generator and RTF Estimator

The sources used for training of the semi-supervised localization method were white noise signals that were convolved with impulse responses produced by a room impulse response (RIR) generator [Hab10]. For each test simulation, we draw at random a

speech signal from the Free American ST Corpus [Pov18]. The corpus includes speech files that were recorded in silence and in-door using a cellphone. It has 10 speakers, with each speaker having about 350 utterances.

4.2 Details on MRF Parameters

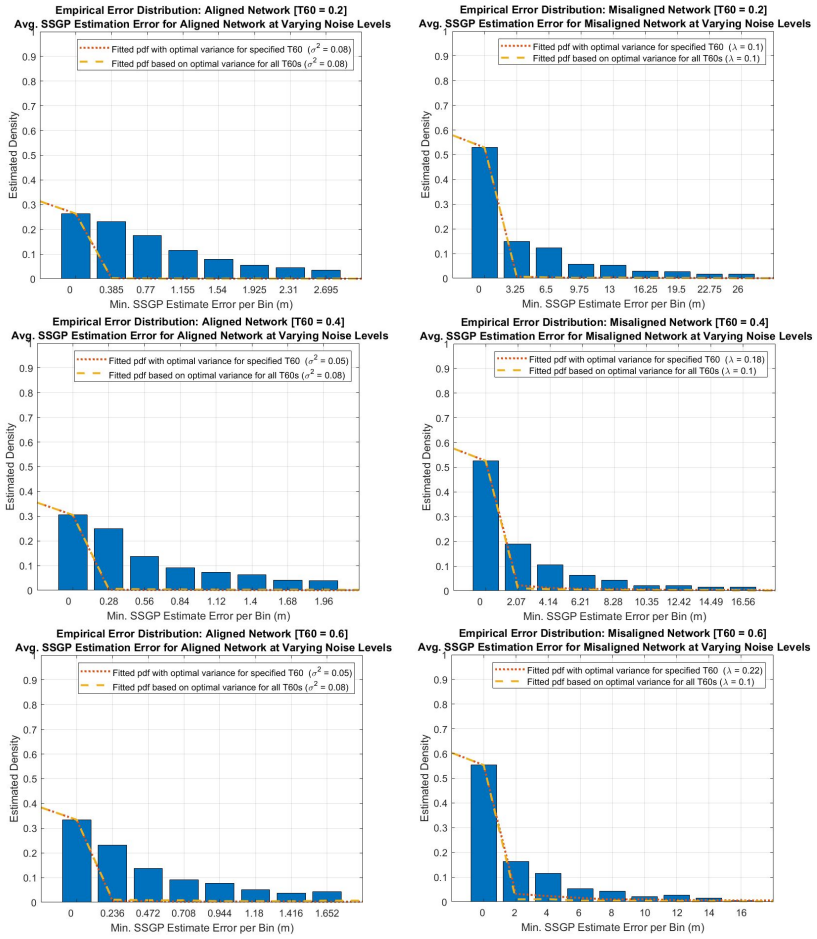


Figure 4.1: Aligned and misaligned latent state error distributions. Distributions are overlaid with PDFs derived using the optimal hyper-parameters (optimal w.r.t. the AUC) for a given T60, and the optimal hyper-parameters where AUC results are averaged over all T60s.

To find the the prior distributions associated with the aligned and misaligned latent classes, we utilize a similar approach to what was done in [AMYHM19], namely we use empirical data. The data was acquired by simulating an audio environment similar

to Fig. 3.2 (with specifications as detailed above) and proceeding as follows: we first estimate a random speech file from a source position for all array sub-networks under 'ideal' conditions (i.e. no additive noise). To estimate the distribution of the aligned class, we then vary the SNR (from 0dB to 30dB by increments of 2dB) and also vary the T60 (values include 0.2, 0.4 and 0.6 s) and estimate the position of a random source and the resulting error between the estimate under 'ideal' conditions and under adverse conditions (with 300 estimates simulated per choice of SNR and T60). Because we are considering the aligned class error distribution, all arrays are in the same position as they were during training. We then plot the localization differences against their relative frequency (normalized such that the output is a probability) and can see there is a correspondence between the localization error distribution with that of the normal distribution (left panels of Fig. 4.1 with each panel displaying the distribution for a given T60). Note that bins are set by ensuring uniform distance between bin values with respect to the maximum and minimum of the observed data. To estimate the distribution of the misaligned class, we perform a similar process, but instead of the arrays remaining static for each different combination of SNR and T60, we randomly choose an array, and shift it 3 meters in a random direction with random rotation of the microphones that comprise the array. In the right panels of Fig. 4.1, we can see that the distribution of error is close to that of an exponential distribution. Lastly, we assume the uncertain class distribution to be uniform as we know that measurements output from the SSGP estimator that are attributed to the uncertain class (on account of the variance inherent to GPs), do not provide information regarding the other latent classes. Note that this was the same assumption made in [AMYHM19].

4.2.1 MRF Hyper-Parameter Optimization

We now consider how the potential function ψ_c, and the hyper-parameters of the priors were obtained. For our considerations the potential function implies the probability of belonging to a given latent state with 3^M different possible configurations of probabili-

ties. In order to optimize the function, we apply the IPFP algorithm whereby we look for $\hat{\psi}_c$ and with the empirical probability of our observations, $\hat{p}_{emp} = \mathbb{P}(\mathbf{e})$, obtained for both the aligned and misaligned class based off the relative distributions generated in the preceding section. This can be seen in 4.2 where the misaligned distribution is binned according to the possible error outcomes related to the aligned experiment and both distributions are normalized with respect to outcomes from both experiments (i.e. the sum of aligned and misaligned y-axis values equals one for a given bin). The hyper-parameters associated with the latent priors, (σ^2_{align}, and λ) were obtained via an ML estimation.

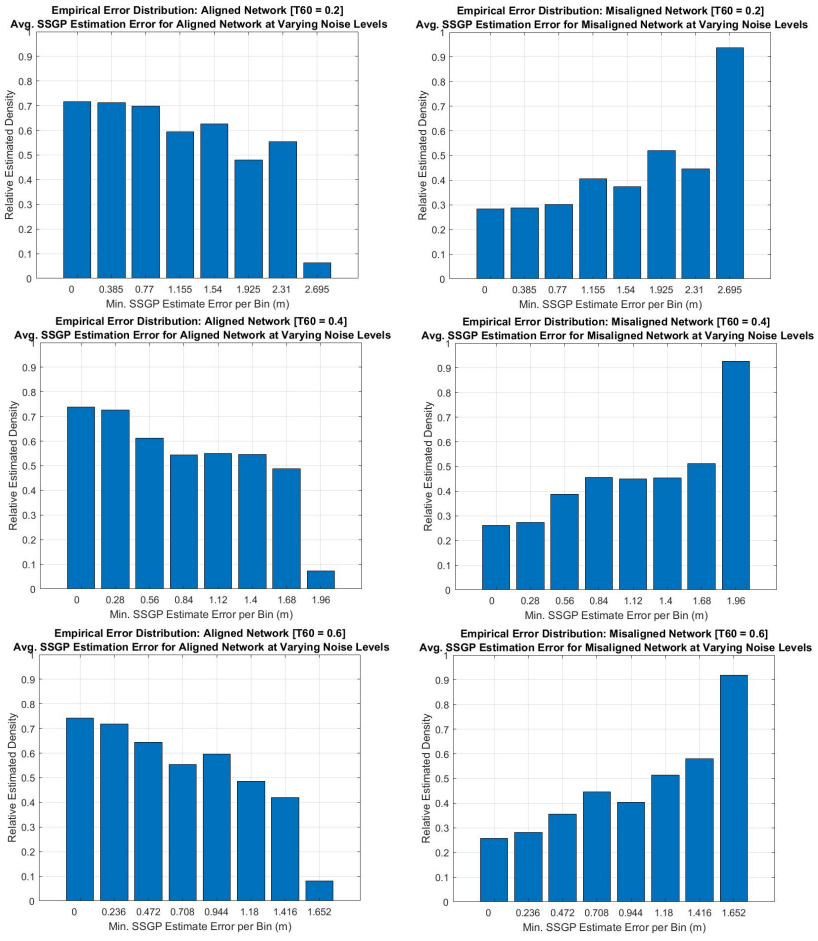


Figure 4.2: Aligned and misaligned latent state error distributions with minimum bin values set with respect to possible aligned error outcomes and distributions normalized with respect to both aligned and misaligned outcomes.

As noted in [Bis06], the potential function is chosen in order to reflect configurations of local variables which are preferred to others specific to the problem at hand. Thus we mention here some of the constraints we enforced in the optimization process. Firstly, we assumed a homogeneous potential function (as did the authors in [AMYHM19]) as the potential of a given sub-network belonging to a given class does not depend on the specific sub-network being considered for the room setup we chose (e.g. no obstructions, and arrays were all uniformly spaced). In future work, the function could be specified for each sub-network to account for microphone array networks that are not uniformly spaced or to account for a room with non-uniform shape. Another assumption we made was if observations belong to the aligned or misaligned class respectively, then they can't simultaneously belong to another. Hence, the corresponding potential probabilities do not sum to one column-wise and thus the potential matrix is not uniform. Moreover the IPFP is only performed row wise and the matrix is not symmetric in general. Below we define the general form of $\hat{\psi}_c$:

$$\hat{\psi}_c = \begin{bmatrix} \hat{p}_{emp} & 0 & \left(1-\hat{p}_{emp}\right) \\ 0 & \hat{p}_{emp} & \left(1-\hat{p}_{emp}\right) \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \tag{4.1}$$

where $\hat{p}_{emp}$ is an IPFP fitted probability based on the empirically driven aligned and misaligned experiments. Specifically, to obtain $\hat{p}_{emp}$ we simulate shifts of a random array (shifts include 0 - 1 m by increments of 0.25 m) and estimate the position of a random sound source for each LONO sub-network. Then based off of the error in estimation, we estimate the probability of the error implying alignment, $p_{emp,a}$ (misalignment, $p_{emp,m}$), in the network from the aligned (misaligned) empirical distributions referenced in 4.2. The IPFP algorithm is applied to the matrix:

$$\psi_c = \begin{bmatrix} p_{emp,a} & \left(1-p_{emp,a}\right) \\ p_{emp,m} & \left(1-p_{emp,m}\right) \end{bmatrix}. \tag{4.2}$$

Thus to obtain the variable components of $\hat{\psi}_c$ in 4.1, we conduct the IPFP algorithm:

Algorithm 1: IPFP

Initialize ψ^t as a 2×2 all-ones matrix with t=1;

while $\sum_{i=1}^{2} \psi_{i,j} \neq \hat{c}_{marginal}$ *or* $\sum_{j=1}^{2} \psi_{i,j} \neq \hat{r}_{marginal}$ **do**

  if $t = 1$ **then**

    $c_{marginal} = \hat{c}_{marginal}$

    $r_{marginal} = \hat{r}_{marginal}$

  else

    $c_{marginal} = \sum_{i=1}^{2} \psi_{i,j}$

    $r_{marginal} = \sum_{j=1}^{2} \psi_{i,j}$

  end

  if t *is odd* **then**

    $\psi_{i,j} = c_{marginal} \odot \frac{\psi_{i,j}}{\sum_{j=1}^{2} \psi_{i,j}}$

  else

    $\psi_{i,j} = r_{marginal} \odot \frac{\psi_{i,j}}{\sum_{j=1}^{2} \psi_{i,j}}$

  end

  $t = t + 1$

end

.

where we denote $\hat{c}_{marginal}$ as the column marginals and $\hat{r}_{marginal}$ as the row marginals of ψ_c for row and column indices i, j respectively. We thus construct $\hat{\psi}_c$ as follows:

$$\hat{\psi}_c = \begin{bmatrix} \psi_{1,1} & 0 & \psi_{1,2} \\ 0 & \psi_{2,1} & \psi_{2,2} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix} \tag{4.3}$$

Note that all parameters were optimized for different choices of T60s (0.2, 0.4 and 0.6 s). In addition, for the training process we use a limited number of speech files from the speech corpus [Pov18] for training.

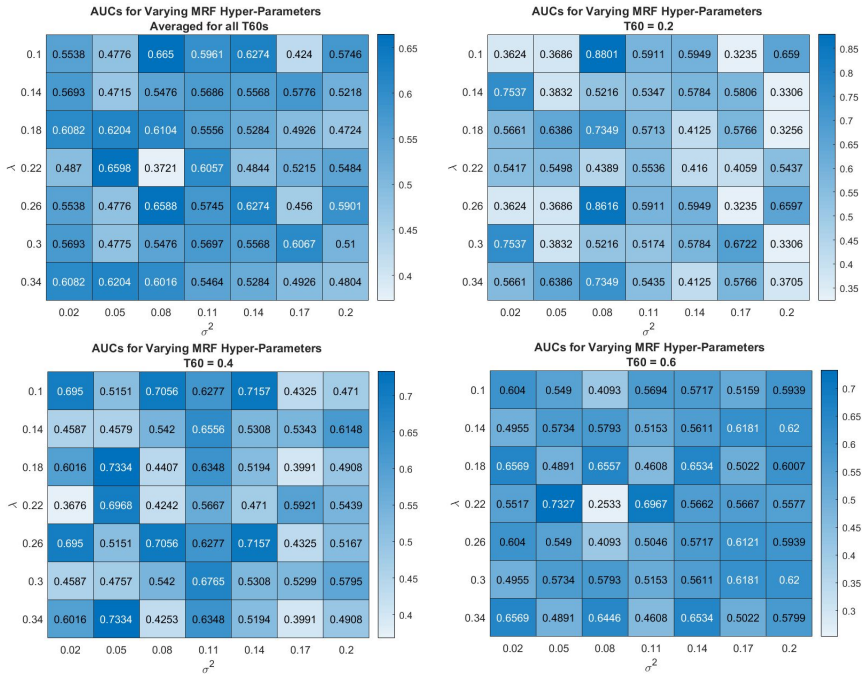
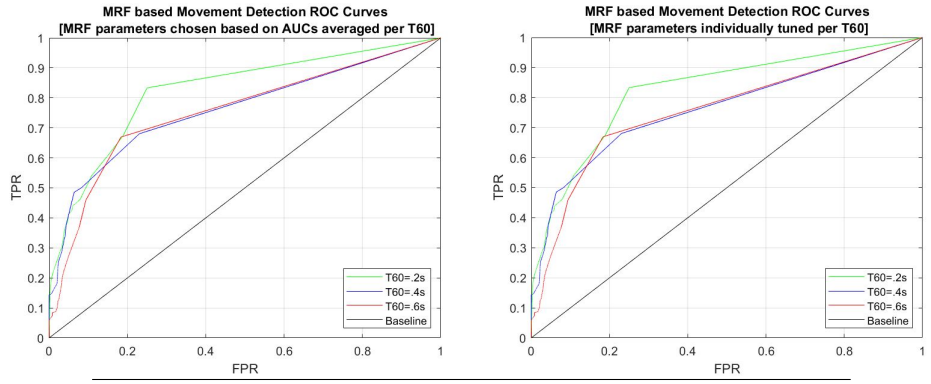


Figure 4.3: AUCs for the MRF-based movement detector with varying prior hyper parameters and T60s (top right, bottom left and bottom right panels), and mean results across all T60s (top left panel).

Looking at Fig. 4.3, we can see how the the detector performs for varying choices of parameters and for different T60s (top right, bottom left and bottom right panels). We also consider which parameters are optimal if we average the AUC results over all T60s. In Fig. 4.1 we overlay the empirical frequencies with the PDFs generated by the optimal choice of hyper-parameters to compare how well the models fit the empirical distributions.

In Fig. 4.4 we can see the difference in performance for the detector using parameters specified for each T60, compared to the case where the parameters were chosen based off the set of average AUCs (averaged over all T60s). The ground truth used was model dependent meaning it was based off whether the error in position estimate after movement was greater than SSGP model mean plus a standard deviation). Also, the total database of audio files is used. It is apparent that specifying the parameters per acoustic situation is not consistently advantageous (case where T60 is 0.6 s). The case where the T60 is 0.2s in fact yields roughly the same AUC as the parameters are the



	T60 = 0.2 s	**T60 = 0.4 s**	**T60 = 0.6 s**
AVG	0.82	0.76	0.76
IND	0.83	0.81	0.74

Figure 4.4: ROCs and corresponding AUCs for MRF-based array movement detection. We consider cases where MRF hyper-parameters are that were optimized via an ML process (optimal w.r.t. the AUC). The left panel shows ROCs for parameters that were chosen w.r.t. average AUC results for all T60s. The right panel reflects results whereby parameters were chosen based off optimal AUC results for each individual T60. The ground truth used was model dependent (i.e. based off whether the error in position estimate after movement was greater than SSGP model mean plus standard deviation).

same.

4.3 Results

As we noted in the introduction, the error in estimation after a single node moves, can yield a significant error in source estimation (even relative to the mean error of the SSGP estimator). In addition, the inconsistency of the error in positional estimates before and after movement indicate the necessity of a more reliable approach (Fig. 1.1). Though there doesn't exist in literature any prevalent estimates for movement in a network of microphone nodes, we consider here some naive estimators. These include a single node estimation comparison technique where we compare estimates of a static sound source position by each individual node in the network to the average of the other three nodes and say if the difference of one estimate is above some threshold,

then movement likely occurred. We also consider a comparison of all LONO sub-networks. Here, we say if the difference in estimation between the LONO sub-networks to the average estimate of the others is above some threshold then movement occurred (note that this is essentially the input to the MRF model). In Fig. 4.5 we see the maximum difference in estimation between a single node and the average of the other three for varying shifts of a random array (left panel) and the difference in estimation of LONO sub-networks for varying shift distances of a random array (center panel).

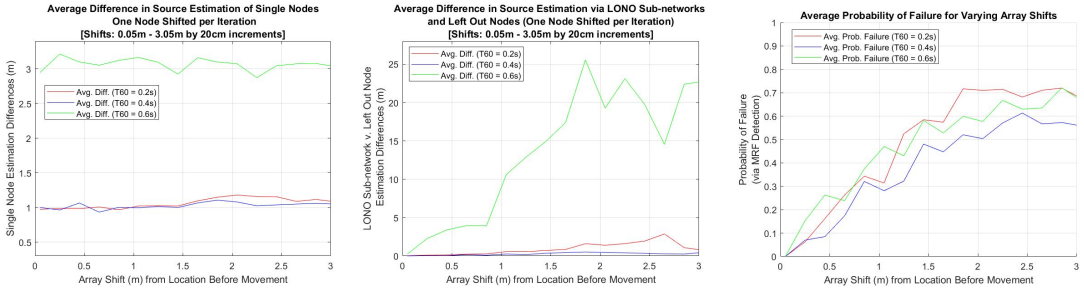
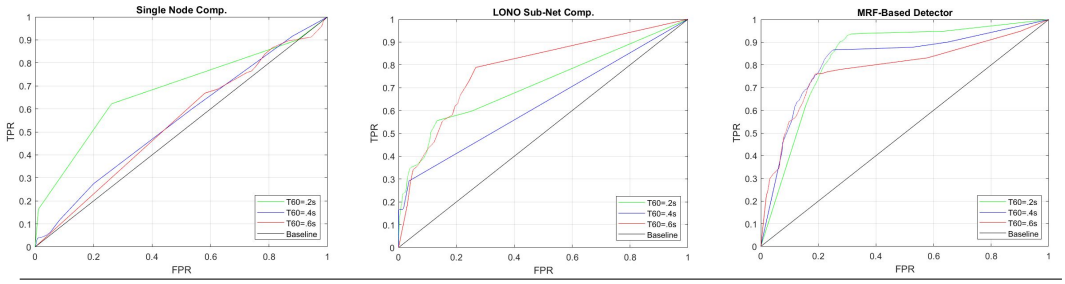


Figure 4.5: Maximum difference in position estimation comparing single node estimates to average estimates of other three single node estimates (left panel), difference in position estimation of LONO sub-network estimates to average estimates of others (center panel) and probability of failure output by the MRF detection algorithm (right panel).

Note that there is not much difference in reported error for varying array shift distances when considering the naive single node comparison approach (left panel). Also, though there is an overall positive correlation between the sub-network naive approach and the shift distance of an array for most cases of T60 (exception being T60 of 0.4 s), the LONO difference in error is highly variable and not consistent. E.g., when the T60 is 0.6 s the range of error is much greater than in other cases. Also, note for T60 cases of 0.2 and 0.6 s, the average difference in error for larger shifts no longer correspond to the shift distance of an array. Both of these qualities of the sub-network naive estimate make it difficult to threshold. On the right most panel we see the result of our proposed method again for comparison. We can see here how incorporating the prior information regarding the error distributions yields a much more stable estimator for movement. It can also be noted that the MRF detection algorithm gives more flexibility in choosing



	T60 = 0.2 s	**T60 = 0.4 s**	**T60 = 0.6 s**
Single Node Comp.	0.68	0.58	0.53
LONO Sub-Net Comp.	0.71	0.62	0.78
MRF-Based Detection	0.84	0.82	0.78

Figure 4.6: ROC curve and corresponding AUCs (table) for different T60 choices (averaged over different ground truth choices [0.1, 0.65, 0.85 and 1.5m]) for the single node position estimation comparison to the average of the other nodes (left panel), LONO sub-network estimation comparison (center panel), and the MRF-based detection method (right panel).

how sensitive a network is to movement (based on the choice of probability threshold).

In Fig. 4.6 we see how the detection algorithms compare with respect to the ROC curve and their corresponding AUCs. To obtain the ROC curves we vary over different thresholds and ground truths and average the results. Thresholds for the naive models were chosen based off the average error of single node and LONO sub-network estimates plus and minus the standard deviation of errors. The specific ground truths were chosen to ensure there was an even distribution of possible outcomes (true and false scenarios) with respect to the array shift distances chosen. E.g. with a ground truth of 0.1 m, and array shifts 0, 0.75, and 1.5 m, there is one case where no movement should be detected (when the shift of the random array is 0 m) and three when movement should be detected. Likewise, if the ground truth is 1.45 m, shifts of 0, and 0.75 m should illicit an output from a given detection model reporting no movement, but a shift of 1.5 m should yield an output indicating movement.

From the table in Fig. 4.6 we can see that the MRF-based detector performs better than the naive estimators across all T60 levels (with respect to the AUC). It is

particularly insightful to look at the center panel (related to the LONO sub-network estimate) and the right panel (related to the MRF detector) and note the improvement achieved using the MRF model and incorporating the prior information regarding the error distributions. In addition, regarding the LONO sub-network naive estimate, one can note the inconsistency in the AUCs and corresponding ROC curves for increasing T60s. This in large part has to do with the difficulty mentioned in choosing a range of thresholds due to the variability in the estimates for increasing shifts (as seen in Fig. 4.5).

Chapter 5

Conclusions

Network integrity is an essential consideration for learning-based localization techniques. In this report, we proposed a method for consistently identifying situations in which network integrity may come into question, namely, when a microphone node in a network has moved. We did so by introducing an algorithm that can probabilistically asses whether a network of nodes is aligned using an MRF model, and based off sub-network sound source estimates localized via a semi-supervised source localization technique. The benefit of the MRF model was shown in comparison to naive estimates that rely directly on relative changes in positional estimates. In particular, the MRF-based detector outputs an indicator that scales commensurate with the size of disruption in the network (disruption in our case meaning the amount a given array moves). This is made possible, as our approach leverage the prior probability of each sub-network's potential latent class (i.e. whether the sub-network is aligned or misaligned) in combination with observed data; in our case the change of position estimate measured by a single LONO sub-network relative to the other sets of sub-networks.

In the future much work can be done to further explore the scope and limitations of the proposed approach. Some topics that can be further explored include utilizing the MRF model for detecting movement of multiple arrays, applying the method to different types of possible situations that may lead to sub-optimal position estimates (e.g. an array in the network has low battery or an array with defective microphones).

One can speculate that as long as prior distributions are modeled in such a way that captures the true frequency of alignment and misalignment, and if changes in positional estimate occur as an array's signal quality deteriorates, then the method should still be applicable. In addition, we noted the algorithm as of now assumes a static source. It would be interesting to see if the node movement detection algorithm could be used for a moving sound source. Finally, there is still the question of what to do regarding the network once detection occurs. Specifically, for the semi-supervised method we considered here, how to go about updating the kernel covariance matrix.

Appendix A

Markov Properties

Given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where V is a set of nodes and $\mathcal{E}$ is a set of edges, a set of random variables $\mathcal{X} = (\mathcal{X}_v)_{v \in \mathcal{V}}$ indexed by $\mathcal{V}$ is an MRF with respect to $\mathcal{G}$ if they satisfy the local Markov properties [Lau96]:

Pairwise Markov property: Any two non-adjacent nodes (i.e. nodes not connected by a vertex), u and v, are conditionally independent given all other nodes:

$$\mathcal{X}_u \perp\!\!\!\perp \mathcal{X}_v \mid \mathcal{X}_{\mathcal{V} \setminus \{u,v\}}. \tag{A.1}$$

Local Markov property: A node, v, is conditionally independent of all other nodes given its neighbors:

$$\mathcal{X} \perp\!\!\!\perp \mathcal{X}_{V \setminus N[v]} \mid \mathcal{X}_{N(v)} \tag{A.2}$$

where N(v) is the set of adjacent nodes (i.e. neighbors) of v, and $N[v] = v \cup \mathcal{N}(v)$ is the closed neighborhood of v.

Global Markov property: Any two subsets of variables are conditionally indepen-

dent given a separating subset:

$$\mathcal{X}_A \perp\!\!\!\perp \mathcal{X}_B \mid \mathcal{X}_S \tag{A.3}$$

where every path from a node in A to a node in B passes through S.

Bibliography

[AMYHM19] Akai, N. ; Morales Yoichi, L. ; Hirayama, T. ; Murase, H.: Misalignment Recognition Using Markov Random Fields with Fully Connected Latent Variables for Detecting Localization Failures. In: *IEEE Robotics and Automation Letters* 4 (Jul, 2019), Nr. 4, S. 3955–3962

[Bis06] Bishop, C. M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg : Springer-Verlag, 2006. – ISBN 0387310738

[BVZ01] Boykov, Yuri ; Veksler, Olga ; Zabih, Ramin: Fast Approximate Energy Minimization via Graph Cuts. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (2001), November, Nr. 11, 1222–1239. http://dx.doi.org/10.1109/34.969114. – DOI 10.1109/34.969114. – ISSN 0162–8828

[CPRD16] Calandra, R. ; Peters, J. ; Rasmussen, C. E. ; Deisenroth, M. P.: Manifold Gaussian Processes for Regression. In: *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016. – ISSN 2161–4407, S. 3338–3345

[DFH13] Deleforge, A. ; Forbes, F. ; Horaud, R.: Variational EM for Binaural Sound-Source Separation and Localization. In: *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.* (2013), S. 76–80

[DFH15] Deleforge, A. ; Forbes, F. ; Horaud, R.: Acoustic Space Learning for Sound-Source Separation and Localization on Binaural Manifolds. In: *Int. J. Neural Syst.* 25 (2015), Nr. 1

[DH12] Deleforge, A. ; Horaud, R.: 2D Sound-Source Localization on the Binaural Manifold. In: *Proc. IEEE Int. Workshop Mach. Learn. Signal Process.* (Sep, 2012), S. 1–6

[ER04] Edward Rasmussen, C.: Gaussian Processes in Machine Learning. In: *Advanced Lectures on Machine Learning* 3176 (2004), S. 63–71

[FM81] Fienberg, S. E. ; Meyer, M. M.: Iterative Proportional Fitting / Department of Statistics, Carnegie-Mellon University. 1981 (270). – Forschungsbericht

[Hab10] Habets, E. ; International Audio Laboratories (Hrsg.): *Room Impulse Response Generator*. Am Wolfsmantel 33, 91058 Erlangen, Germany: International Audio Laboratories, Sep, 2010. – https://github.com/ehabets/RIR-Generator

[HC71] Hammersley, J.M. ; Clifford, P.: *Markov Fields on Finite Graphs and Lattices*. 1971

[KB19] Kato, K ; Brandão, G. S. L.: Quantum Approximate Markov Chains are Thermal", journal="Communications in Mathematical Physics. 370 (2019), Aug, Nr. 1, 117–149. http://dx.doi.org/10.1007/s00220-019-03485-6. – DOI 10.1007/s00220–019–03485–6. – ISSN 1432–0916

[KC76] Knapp, C. ; Carter, G.: The Generalized Correlation Method for Estimation of Time Delay. In: *IEEE Trans. Antennas Propag.* ASSP-24 (Aug, 1976), Nr. 4, S. 320–327

[Lau96] Lauritzen, S.: *Graphical Models*. Clarendon Press, 1996

[LGTG16] Laufer-Goldshtein, B. ; Talmon, R. ; Gannot, S.: Semi-Supervised Sound Source Based on Manifold Regularization. In: *IEEE/ACM Transactions on Audio, Speech and Language Processing* 24 (Aug, 2016), Nr. 8, S. 1393–1407

[LGTG17a] Laufer-Goldshtein, B. ; Talmon, R. ; Gannot, S.: Speaker Tracking on Multiple-Manifolds with Distributed Microphones. In: *Latent Variable Analysis and Signal Separation* 10169 (Feb, 2017), S. 59–67

[LGTG17b] Laufer-Goldshtein, B. ; Talmon, R. ; Gannot, S.: Semi-Supervised Source Localization on Multiple-Manifolds with Distributed Microphones. In: *IEEE/ACM Transactions on Audio, Speech and Language Processing* 25 (Jul, 2017), Nr. 7, S. 1477–1491

[Li95] Li, S.Z.: Markov Random Field Modeling in Computer Vision. In: *Springer, Tokyo* (1995)

[LM01] Lafferty, J.D. ; Mccallum, A.: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *Proceedings of the Eighteenth International Conference on Machine Learning* (2001), S. 282–289

[LTG13] Laufer, B. ; Talmon, R. ; Gannot, S.: Relative Transfer Function Modeling for Supervised Source Localization. In: *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, S. 1–4

[MPK11] May, T. ; Par, S. van d. ; Kohlrausch, A.: A Probabilistic Model for Robust Localization Based on a Binaural Auditory Front-End. In: *IEEE Trans. Audio, Speech, Lang. Process.* 19 (Jan, 2011), Nr. 1, S. 1–13

[Mur12] Murphy, K. P.: *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. – ISBN 0262018020

[Pov18] Povey, D.: *"ST-AEDS-20180100_1, Free ST American English Corpus"*. `https://www.openslr.org/45/`. Version: 2018

[RK89] Roy, R. ; Kailath, T.: ESPRIT-Estimation of Signal Parameters via Rotational Invariance Techniques. In: *IEEE Int. Conf. Acoust., Speech, Signal Process.* 37 (Jul, 1989), Nr. 7, S. 984–995

[Sch86] Schmidt, R. O.: Multiple Emitter Location and Signal Parameter Estimation. In: *IEEE Trans. Antennas Propag.* AP-34 (Mar, 1986), Nr. 3, S. 276–280

[SCK07] Sindhwani, W. ; Chu, W. ; Keerthi, S.S.: Semi-supervised Gaussian Process Classifiers. In: *Proc. 20th Int. Joint Conf. Artif. Intell.* (2007), S. 1059–1064

[TBF05] Thrun, S. ; Burgard, W. ; Fox, D.: *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. – ISBN 0262201623

[TCGC13] Talmon, R. ; Cohen, I. ; Gannot, S. ; Coifman, R.R.: Diffusion Maps for Signal Processing. In: *IEEE Signal Processing Magazine* (Jul, 2013), S. 75–86

[YCH02] Yao, K. ; Chen, J. C. ; Hudson, R. E.: Maximum-Likelihood Acoustic Source Localization: Experimental Results. In: *IEEE Int. Conf. Acoust., Speech, Signal Process.* 3 (2002), S. 2949–2952