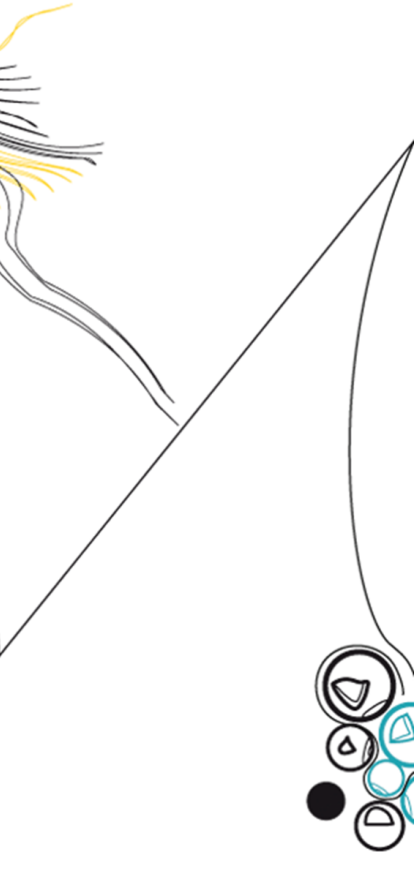
UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**

Direction-of-arrival estimation of an unknown number of signals using a machine learning framework

Noud B. Kanters

M.Sc. Thesis
January 2020



Supervisors:
dr. A. Alayón Glazunov
dr. ir. A. B. J. Kokkeler
dr. ing. E. A. M. klumperink
dr C. G. Zeinstra

Telecommunication Engineering Group
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Preface

This thesis has been written as the final part of my master's programme Electrical Engineering at the university of Twente. The research presented in this document has been conducted within the Telecommunication Engineering chair and serves as their first investigation into the field of direction-of-arrival estimation aided by machine learning.

I would like to thank my main supervisor Andrés Alayón Glazunov for the valuable discussions we had throughout the entire period of the assignment, as well as for his constructive feedback on this thesis. Furthermore, I would like to thank Chris Zeinstra for his input regarding the machine learning component of the work, an area completely new to me. Finally, I would like to express my gratitude to the members of the committee who assessed this thesis for their time.

Summary

Direction-of-arrival (DOA) estimation is a well-known problem in the field of array signal processing with applications in, e.g., radar, sonar and mobile communications. Many conventional DOA estimation algorithms require prior knowledge about the source number, which is often not available in practical situations. Another common feature of many DOA estimators is that they aim to derive an inverse of the mapping between the sources' positions in space and the array output. However, in general this mapping is incomplete due to unforeseen effects such as array imperfections. This degrades the performance of the DOA estimators.

In this work, a machine learning (ML) framework is proposed which estimates the DOAs of waves impinging an antenna array, without any prior knowledge about the number of sources. The inverse mapping mentioned above is made up by an ensemble of single-label classifiers, trained on labeled data by means of supervised learning. Each classifier in the ensemble analyses a number of segments of the discretized spatial domain. Their predictions are combined into a spatial spectrum, after which a peak detection algorithm is applied to estimate the DOAs.

The framework is evaluated in combination with feedforward neural networks, trained on synthetically generated data. The antenna array is a uniform linear array of 8 elements with half wavelength element spacing. A framework with a grid resolution of 2°, trained on 10^5 observations of 100 snapshots each, achieved an accuracy of 93% regarding the source number for signal-to-noise ratios (SNRs) of at least -5 dB when 2 uncorrelated signals impinge the array. The root-mean-square error (RMSE) of the estimates of the DOAs of these observations is below 1° and equals 0.5° for SNRs of 5 dB and higher. It is shown that in the remaining 7%, the DOAs are spaced 2.4° degree on average, making the resolution of the grid too coarse for resolving these DOAs.

Increasing the resolution of the grid is at the cost of an increased class imbalance, which complicates the classification procedure. Nevertheless, it is shown that a 100% probability of resolution is obtained for observations of 15 dB SNR with DOA spacings of at least 3.2° for a framework of 0.8° resolution, whereas the framework of 2° resolution achieves this for spacings larger than 5.9°. However, 4 times more training data is used to realize this.

A scenario with a variable source number showed that the performance of the ML framework decreases gradually with an increasing number of sources. When a single signal with a 15 dB SNR impinges the array, this is estimated correctly in 100.0% of the observations, with an RMSE of 0.4°. However, when 7 sources exist, the performance decreases to 3.3% and 1.8° respectively. A decreased accuracy of the source number estimates was expected because of the 2° resolution that was used. However, it is shown that the performance of the neural networks in terms of their predictions decreases with an increasing source number as well.

The results indicate that the resolution of the framework has a significant impact on its DOA estimates. It is observed that for the considered learning strategy, additional training data is required to actually benefit from an increased resolution. Further research is required to determine if alternative learning algorithms and advanced techniques for handling class imbalance could diminish this need for additional data. Furthermore, it should be verified if the proposed data-driven approach indeed adapts better to unforeseen effects compared to model-based algorithms by evaluating it on real-world data.

Contents

List of acronyms

AIC	Akaike information criterion
AOA	angle-of-arrival
BR	binary relevance
BWNN	null-to-null beamwidth
CRLB	Cramér-Rao lower bound
DNN	deep neural network
DOA	direction-of-arrival
ESPRIT	estimation of signal parameters via rotational invariance techniques
FFNN	feedforward neural network
i.i.d.	independent and identically distibuted
LOS	line-of-sight
LP	label powerset
MDL	minimum description length
ML	machine learning
MSE	mean-square error
MUSIC	multiple signal classification
NN	neural network
RAkEL	random k-labelsets
ReLU	rectified linear unit
rms	root-mean-square

RMSE root-mean-square error

SNR signal-to-noise ratio

SVR support vector regression

ULA uniform linear array

Chapter 1

Introduction

Estimating the direction-of-arrival (DOA), or angle-of-arrival (AOA), of multiple waves impinging a sensor array is a well-known problem in the field of array signal processing. It has a wide range of applications in, for example, radar, sonar and mobile communications. In practical situations, the number of sources is often unknown to the estimator, complicating the estimation process.

The DOA estimation problem has been addressed by, e.g., the popular subspace-based superresolution methods multiple signal classification (MUSIC) [1] and estimation of signal parameters via rotational invariance techniques (ESPRIT) [2]. However, both methods require prior knowledge about the source number. With an increasing amount of computational power being available, sparsity-based approaches have become popular as well [3]. A common feature of the techniques mentioned above is that they rely on a model which maps the sources' positions in space to the signals received by the sensor array. The DOA estimation is essentially a matter of finding the inverse of this mapping. However, in practice the forward mapping will contain imperfections because of, e.g., array imperfections, modelling errors in the sensors' transfer functions, mutual coupling between the elements and the presence of noise. This will affect the inverse mapping as well, and with that degrade the performance of the DOA estimation algorithms.

As an alternative to computing an inverse mapping based on the (most likely) incomplete forward mapping, one could derive the inverse mapping directly from labeled input-output pairs, i.e. from real array outputs of which the corresponding source positions are known. As a result, factors such as array imperfections and the sensors' transfer functions are included implicitly. This approach is called supervised learning, a well-known branch of machine learning (ML). This technique is the core of the assigment addressed in this thesis.

1.1 Goals of the assignment

The main goal of the research presented in this thesis is summarized in the following statement:

Devise a machine learning framework which is able to estimate the directions-of-arrival of an unknown number of signals.

The idea behind this assignment is to find out the advantages, if any, of utilizing ML to solve this well-known DOA estimation problem. The work is not related to a particular application, meaning that no exact performance criteria are specified. Furthermore, no requirements regarding the ML algorithm or the antenna array are given. Ideally, the framework is constructed in a way that it can be employed in combination with any array configuration, such that it can be applied to both 1D and 2D DOA estimation.

1.2 Related work

Two well-known DOA estimation algorithms, both mentioned above, are MUSIC [1] and ESPRIT [2]. Whereas MUSIC is based on the noise subspace, ESPRIT employs the signal subspace. Both methods are by definition limited to estimating the DOAs of at most $N - 1$ signals, with N being the number array elements. The number of signals must be known before being able to estimate the DOAs. If the latter does not apply, it is to be estimated using, e.g., a subspace order estimator like the minimum description length (MDL) or Akaike information criterion (AIC) [4].

A. Khan et al. [5] combined the MUSIC algorithm [1] with several ML techniques for the 2D DOA estimation of a single target. It was shown that the DOA estimation performance in terms of mean absolute error improved aided by ML compared to using the MUSIC algorithm on its own. However, none of the considered ML techniques clearly outperformed the others.

In [6], 1D DOA estimation of two equally powered, uncorrelated sources using a deep neural network (DNN) was investigated. The DNN acts as a classifier with a 1° resolution and uses the estimated sensor covariance matrix of a 5-element uniform linear array (ULA) as an input. Only integer DOAs were considered. For a signal-to-noise ratio (SNR) of 30 dB, the estimation error was within 1° in 97% of the observations.

Z. Liu et al. [7] approached the 1D DOA estimation problem using a DNN as well. The DNN consists of a multitask autoencoder and a number of parallel multilayer classifiers. In the case of two unequally powered sources (10 and 13 dB SNR) separated by 16.4°, the estimation errors for both signals were kept within 1°,

whereas the support vector regression (SVR) method proposed in [8] resulted in errors upto 5° for the same scenario. The DNN was trained on a dataset consisting of 10 dB SNR observations only.

O. Bialer et al. [9] combined classification and regression in a single DNN. The neurons of the classifying part predict the number of sources, which is assumed to be between 1 and 4. Based on this prediction, a particular set of regression neurons containing the DOA estimates is to be read out. For a single snapshot, an SNR of 40 dB and an ULA of 16 elements, the probability that the number of sources is estimated correctly equals 90%.

1.3 Thesis organization

In Chapter 2, the problem statement is presented by means of the underlying data model. Then, in Chapter 3, the ML framework is introduced and the employed learning algorithm is discussed. In Chapter 4, the simulations that were conducted to assess the performance of the proposed framework are presented. Finally, the thesis is concluded in Chapter 5.

Chapter 2

Problem statement

In this chapter, the problem is formulated by means of a model based on well-known theoretical models presented in, e.g., [5], [7], [10], [11]. The data used for training and testing the ML framework is created using this model as well, as no real-world measurements were conducted within this assignment.

2.1 Data model

Consider K complex-valued narrow-band signals impinging an antenna array consisting of N isotropic elements. The sources transmitting these signals are assumed to be in the far-field of the array and the antenna elements of transmitters and receiver are co-polarized. With $y_n(t)$ being the sample received by the n^{th} element, i.e. $n = 1, \ldots, N$, at the t^{th} time instance, the data vector $\mathbf{y}(t) = [y_1(t), \ldots, y_N(t)]^T$ is modelled as

$$\mathbf{y}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \tag{2.1}$$

where $\mathbf{y}(t) \in \mathbb{C}^N$, $\mathbf{A} \in \mathbb{C}^{N \times K}$ is the array manifold, $\mathbf{s}(t) \in \mathbb{C}^K$ is a vector containing the complex amplitudes of the transmitted signals and $\mathbf{n}(t) \in \mathbb{C}^N$ is a vector containing the additive noise per antenna element.

The array manifold is given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_K \end{bmatrix}, \tag{2.2}$$

where $\mathbf{a}_k \in \mathbb{C}^N$ is the steering vector associated with the k^{th} source, i.e. $k = 1, \ldots, K$. The k^{th} steering vector is given by

$$\mathbf{a}_k = \begin{bmatrix} a_{1,k} & a_{2,k} & \cdots & a_{N,k} \end{bmatrix}^T, \tag{2.3}$$

and depends on the positions of the array elements relative to a reference point, the direction information of the signals, and the wavelength λ. The n^{th} element of the k^{th}

steering vector is defined as

$$a_{n,k} = e^{-j\frac{2\pi}{\lambda}\mathbf{r}_n^T\mathbf{w}_k}. \tag{2.4}$$

The vector $\mathbf{r}_n \in \mathbb{R}^3$ contains the Cartesian coordinates of the n^{th} array element Rx_n relative to the reference point

$$\mathbf{r}_n = \begin{bmatrix} x_n & y_n & z_n \end{bmatrix}^T \tag{2.5}$$

and $\mathbf{w}_k \in \mathbb{R}^3$ is composed of the Cartesian coordinates of the unit-vector pointing from the reference point towards the k^{th} source Tx_k. These Cartesian coordinates are computed from the azimuth angle ϕ_k and the elevation angle θ_k as follows:

$$\mathbf{w}_k = \begin{bmatrix} \cos\theta_k \ \cos\phi_k \\ \cos\theta_k \ \sin\phi_k \\ \sin\theta_k \end{bmatrix}. \tag{2.6}$$

Without loss of generality, it is assumed that $\phi_1 < \cdots < \phi_K$ and $\theta_1 < \cdots < \theta_K$. All geometry related parameters are visualized in Fig. 2.1.

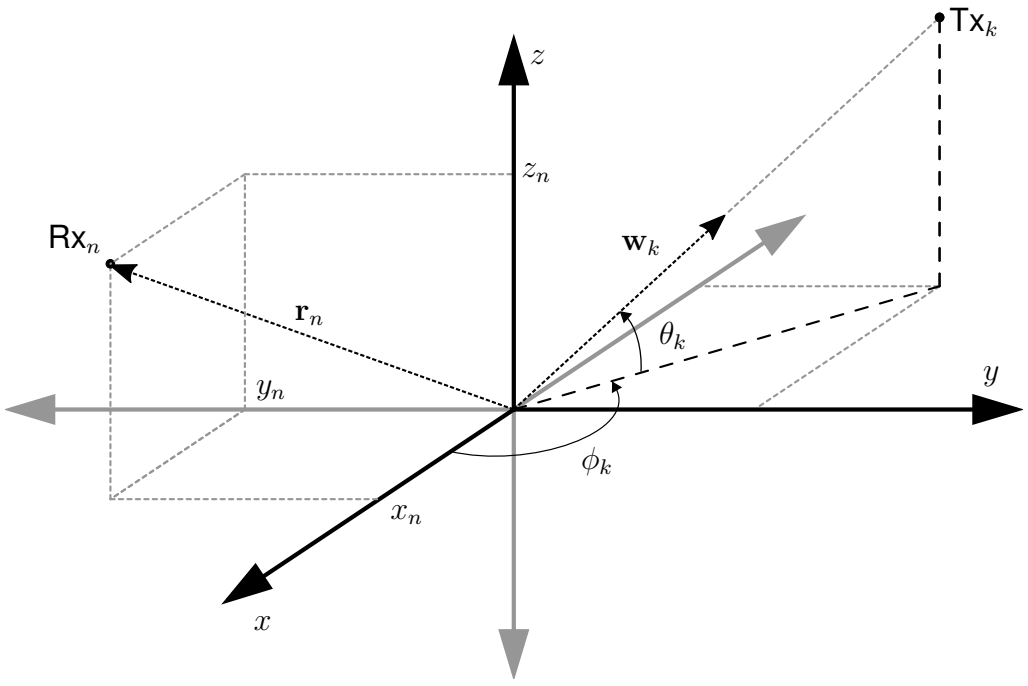


Figure 2.1: Geometry definitions.

When T snapshots are available, i.e. $t = 1, \ldots, T$, equation 2.1 can be written as the matrix equation

$$\mathbf{Y} = \mathbf{AS} + \mathbf{N}, \tag{2.7}$$

with, $\mathbf{Y} = [\mathbf{y}(1), \ldots, \mathbf{y}(T)]$, $\mathbf{S} = [\mathbf{s}(1), \ldots, \mathbf{s}(T)]$ and $\mathbf{N} = [\mathbf{n}(1), \ldots, \mathbf{n}(T)]$.

2.2 Assumptions and conditions

Aided by the model described in section 2.1, the problem can be defined more specifically. The core of the problem is to estimate the DOAs of the K uncorrelated narrow-band signals impinging the array, with K being unknown to the estimator. The 2D DOA of the k^{th} signal is defined by two parameters: the azimuth angle ϕ_k and the elevation angle θ_k. Each of the parameters mentioned above, i.e. K, ϕ_k and θ_k (with $k = 1, \ldots, K$), are assumed to be constant over all T snapshots within a single observation. Furthermore, it is assumed that both $\mathbf{s}(t)$ and $\mathbf{n}(t)$ are independent and identically distibuted (i.i.d.) random variables following complex Gaussian distributions

$$\mathbf{s}(t) \sim \mathcal{CN}(\mathbf{0}, \sigma^2 \mathbf{I}_K) \tag{2.8a}$$

$$\mathbf{n}(t) \sim \mathcal{CN}(\mathbf{0}, \nu^2 \mathbf{I}_N)) \tag{2.8b}$$

with σ^2 being the signal variance, ν^2 the noise variance, and $\mathbf{I}_K$,$\mathbf{I}_N$ identity matrices of size K and N respectively. In other words, the DOA estimator has no knowledge about the signals transmitted by the sources. Furthermore, equation 2.8 implies that all signals within a single observation have the same SNR, i.e. σ^2/ν^2.

The framework developed to solve this DOA estimation problem is presented in Chapter 3.

Chapter 3

Method

Supervised learning algorithms can be roughly divided in two categories: classification algorithms and regression algorithms [12]. The core of the problem considered in this work is the unknown, possibly varying, number of sources K. This implies that the number of target outputs of the framework could differ for various observations. Solving the problem using regression would therefore require a method similar to the one presented in [9], where the number of sources is predicted using a classifier prior to estimating the actual DOAs via regression. However, this implies that the design of the ML framework imposes a limit on the amount of DOAs that can be estimated. It was therefore decided to construct a framework which is solely based on classification and which does not need another algorithm to estimate the number of sources. This is achieved by discretizing the spatial domain, which comes at the cost of a finite estimation resolution. It was decided to consider 1D DOA estimation only, although the data model presented in section 2.1 could be used to generate 2D data as well. The azimuth angles $\phi_1, \ldots, \phi_K$ are to be estimated, whereas the elevation angles $\theta_1, \ldots, \theta_K$ are fixed at 0 degrees. The principles behind the framework can be easily extended to 2D. The framework is presented in sections 3.1 and 3.2, whereas the employed learning algorithm is discussed in section 3.3.

3.1 DOA estimation via classification

The first step towards estimating DOAs via classification is to define a grid. The spatial domain of interest, $[\phi_{min}, \phi_{max}]$ with $\phi_{max} > \phi_{min}$, is divided in M equal segments. The width of one segment, $\Delta\phi$, follows from

$$\Delta\phi = \frac{\phi_{max} - \phi_{min}}{M}, \tag{3.1}$$

for any positive integer M. If the DOA ϕ of a signal impinging the array is associated with the i^{th} segment, $i = 1, \ldots, M$, its DOA estimate $\hat{\phi}$ is the centre of that

segment, c_i. The same procedure is used if K signals impinge the array from angles $\phi_1, \ldots, \phi_K$, as visualised in Fig. 3.1. Note that if multiple DOAs correspond to the same grid segment, they cannot be resolved.

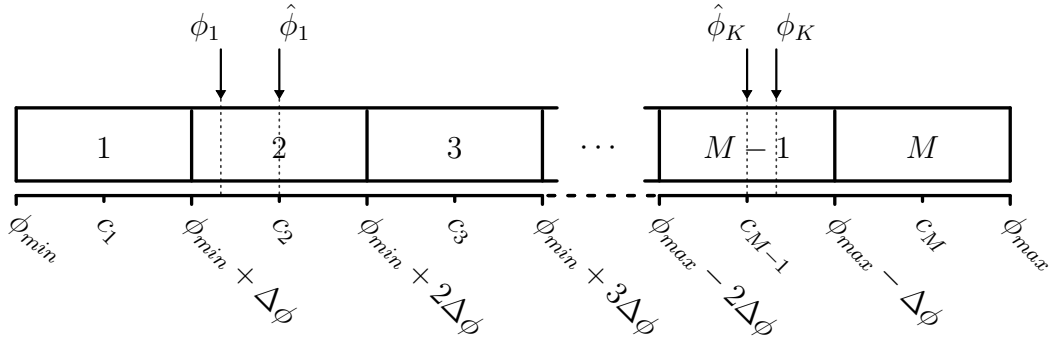


Figure 3.1: DOA estimation via classification.

The approach described above could be implemented using a multi-label multi-class (or simply multi-label) classifier: M classes exist of which K are true for a single observation. In other words, K labels should be assigned. Multi-label learning problems have been investigated thoroughly. An overview of several methods to deal with this kind of ML problems is presented in [13]. A distinction is made between *problem transformation* and *algorithm adaptation* methods. Algorithms in the former category transform the task into a more manageable problem such as binary classification or multi-class (single-label) classification. Techniques in the algorithm adaptation category are adapted versions of well-known ML algorithms, such that they can deal with multi-label data without transforming it.

As the problem statement of this thesis does not put any restriction on the ML algorithm to be used, a framework is proposed which can be combined with any single-label multi-class classification algorithm. In this way, different algorithms could be compared in a later stage. The framework is based on the ensemble method random k-labelsets (RAkEL), proposed in [14]. RAkEL aims to achieve a high classification performance while keeping computational complexity low. Section 3.2 presents how the RAkEL framework is employed to solve the given DOA estimation problem.

3.2 The framework

RAkEL is a framework which can be used to solve a multi-label classification problem using an ensemble of single-label classifiers. Before explaining the details of RAkEL, it is important to understand the concept of a label powerset (LP).

3.2.1 Label powerset

LP is a technique which can be employed to transform a problem from multi-label to single-label [13]. It considers all 2^M combinations of M possible labels. For example, for a multi-label classification problem with 2 labels, the LP consists of $2^2 = 4$ classes. These classes are referred to as (00), (01), (10) and (11), where a 1 indicates that a label is assigned and a 0 denotes the opposite. Each digit represents one label. In this way, a single-label problem is obtained without losing information about possible correlations between the labels of the original multi-label task. The latter does not apply to, e.g., the binary relevance (BR) method, where M single-label classifiers are trained: one for each of the M labels. A disadvantage of LP is that the number of classes grows exponentially with M. This complicates the application of LP for domains with large M as many classes will be represented by few training examples [15]. The latter problem is addressed by RAkEL, as will be shown in the next paragraph.

3.2.2 RA*k*EL

The main principle behind RAkEL [14] is the division of the single-label classification problem of 2^M classes in m smaller problems of 2^k classes, i.e. $k < M$. This is achieved by splitting the original set of M labels in multiple subsets of k labels. These subsets, from now on referred to as labelsets, are generated via random sampling from the original set. Single-label classifiers are trained on the LPs of those labelsets. Each label might or might not appear in multiple labelsets, referred to as RAkEL$_\text{o}$ (overlapping) and RAkEL$_\text{d}$ (disjoint) respectively. In other words, the random sampling can be performed either with or without replacement. For RAkEL$_\text{o}$, the final prediction for each label is obtained via a majority voting procedure over the entire ensemble. An example from [15] with $m = 7$, $M = 6$ and $k = 3$ is presented in Table 3.1. The labels $c_1, \ldots, c_6$ can be considered as being the class centres shown in Fig. 3.1.

In [15], RAkEL is compared to 6 other multi-label learning techniques from both the transformation as well as the adaptation category. It is shown that, averaged over 8 different databases, RAkEL$_\text{o}$ with $k = 3$ and $M < m < 2M$ outperforms the considered techniques. Furthermore, it outperforms RAkEL$_\text{d}$ for 7 of the 8 considered databases.

3.2.3 Modification 1 - combining RA*k*EL$_\text{o}$ and RA*k*EL$_\text{d}$

A disadvantage of RAkEL$_\text{o}$ is the imbalance in the amount of labelsets in which the labels of the original set appear, i.e. the denominators in the 'average votes' row in

Table 3.1: RAkEL$_o$ example [15]

		Predictions					
Classifier	Labelset	c_1	c_2	c_3	c_4	c_5	c_6
1	$\{c_1,c_2,c_6\}$	1	0	-	-	-	1
2	$\{c_2,c_3,c_4\}$	-	1	1	0	-	-
3	$\{c_3,c_5,c_6\}$	-	-	0	-	0	1
4	$\{c_2,c_4,c_5\}$	-	0	-	0	0	-
5	$\{c_1,c_4,c_5\}$	1	-	-	0	1	-
6	$\{c_1,c_2,c_3\}$	1	0	1	-	-	-
7	$\{c_1,c_4,c_6\}$	0	-	-	1	-	0
	Average votes	3/4	1/4	2/3	1/4	1/3	2/3
	Final prediction	1	0	1	0	0	1

Table 3.1. This imbalance is a result of the random sampling and causes variations in the classification accuracy over the different labels: a label which appears in more labelsets will be assigned more accurately than labels covered by less classifiers in general. Furthermore, it could occur that certain labels are not selected at all. In the given DOA estimation application, this implies that specific sections of the spatial domain are not taken into account. This is unwanted, as a geometrically symmetric configuration of the sensors and sources should result in symmetric DOA estimation performance. It was therefore decided to use L 'layers' of RAkEL$_d$ instead of RAkEL$_o$, as is visualized in Fig. 3.2. The labelsets consisting of k labels are defined for each layer individually, as indicated by the shaded blocks.

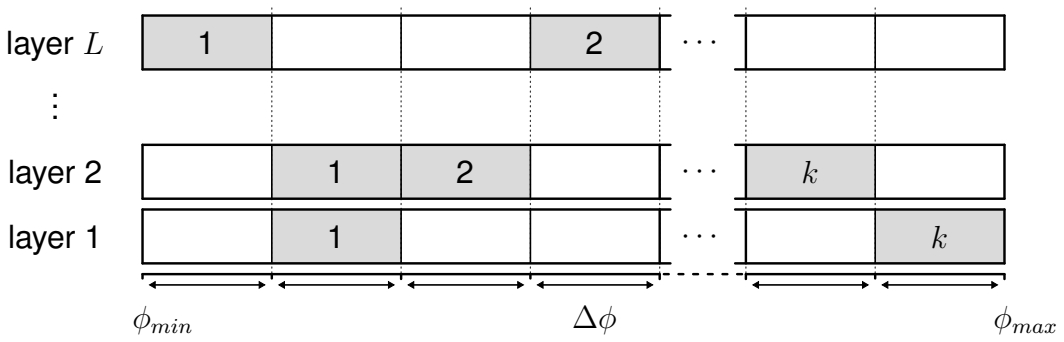


Figure 3.2: DOA estimation framework consisting of multiple layers of RAkEL$_d$.

The total number of classifiers m in the framework follows from the number of layers L, the amount of labels M and the number of labels in a labelset k according to

$$m = L \left\lceil \frac{M}{k} \right\rceil. \tag{3.2}$$

3.2.4 Modification 2 - border perturbations

A disadvantage of the discretization of the spatial domain is that the estimation error $|\phi - \hat{\phi}|$ approaches $\Delta\phi/2$ when ϕ approaches the border between two segments. As an additional result of the modification presented in section 3.2.3, this could be improved by making sure that the borders of different layers appear at different angles. It was therefore decided to perturb the borders for each RAkEL layer individually. An example of what the complete classifier framework could look like is shown in Fig. 3.3.

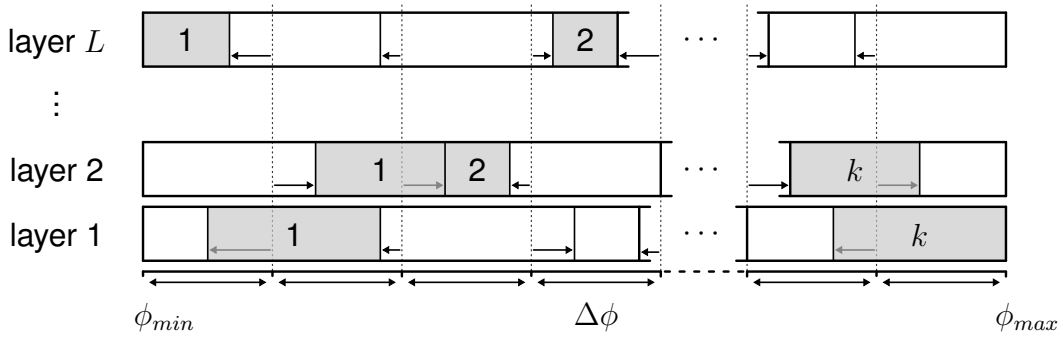


Figure 3.3: DOA estimation framework with perturbed borders.

An artefact of these perturbations is that the DOA estimates can no longer be obtained via the straightforward majority voting procedure shown in Table 3.1. However, the majority voting procedure can also be regarded as the comparison of some spectrum with the value $L/2$. This spectrum appears when summing the estimates of all layers in the framework. This procedure can also be applied after perturbing the borders. The approach described above is illustrated by means of an example of $L = 3$ layers, shown in Fig. 3.4.

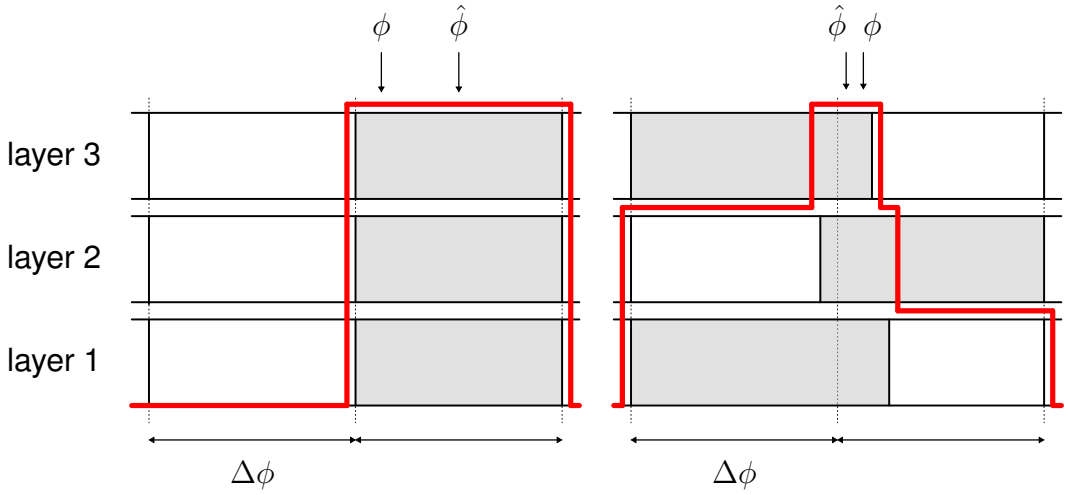


Figure 3.4: DOA estimation without (left) and with (right) border perturbation.

The arrows labeled with 'ϕ' represent a signal impinging the array from an azimuth angle ϕ. Each block in a layer represents a segment of the discretized spatial domain. A shaded block indicates a positive estimate, i.e. the label of that grid segment is assigned to the observation. Note that perfect classifiers are assumed in this example. By summing all estimates over the different layers, a spectrum (indicated by the red lines) appears. It can be seen that the perturbation of the borders (right) results in a DOA estimate $\hat{\phi}$ (the middle of the peak plateau) which is closer to the true DOA ϕ than the estimate that would be obtained without perturbations (left). A more detailed explanation of how the DOA estimates follow from the spectra is presented in section 3.2.5.

3.2.5 Modification 3 - peak detection

In section 3.2.4, it was shown how a spectrum is constructed based on the predictions of the classifier ensemble. The DOA estimates are obtained by applying a peak detection algorithm to this spectrum. This algorithm computes all local maxima and compares them to some threshold. Only the peaks higher than the threshold are returned as being a DOA estimate. A threshold of $L/2$ can be interpreted as the majority voting procedure usually applied in RAkEL$_o$, see Table 3.1. If a peak has a flat top as in the example of Fig. 3.4, the argument of the centre of the plateau is taken as the estimate. The peak detection procedure is visualized in Fig. 3.5.

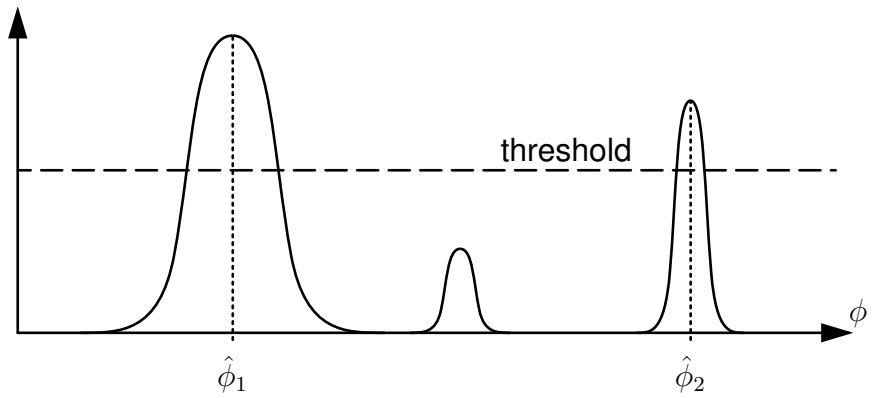


Figure 3.5: Peak detection applied to a spatial spectrum.

Instead of using a fixed threshold, it was decided to optimize it using the data that is available. A set of calibration spectra is obtained by feeding the trained classifier ensemble with observations it never saw before. These spectra, of which the associated parameters K and $\phi_1, \ldots, \phi_K$ are known, can be evaluated using various threshold values. The value which maximizes the amount of observations for which $\hat{K} = K$, with $\hat{K}$ being the estimate of K, is used as a threshold for new observations. In this way, the threshold is adapted to the data.

A downside of this straightforward peak detection procedure is that two signals associated with two neighbouring segments of the grid cannot be resolved as they will result in a single peak. This might be taken into account by considering the width of the peak as well, which is a recommended investigation for the future. For now, two DOAs can only be resolved if their associated grid segments are separated by at least one other segment.

3.3 The learning algorithm

Given the framework presented in section 3.2, a base-level single-label learning algorithm is to be chosen. Examples of such algorithms are decision trees, support vector machines and neural networks. Little literature is available in which the performance of different algorithms in the area of DOA estimation is compared. In [5] such a comparison is presented, but none of the three considered algorithms clearly outperforms the others. Furthermore, the scenario considered there is different, as the algorithms are trained on 2D MUSIC spectra. After all, it was decided to use the well-known feedforward neural network (FFNN) as a base-level algorithm. FFNNs come with a lot of design freedom and much literature has been published about using (deep) FFNNs for DOA estimation in the past few years, e.g. [7], [9], [10]. In spite of that, one of the recommendations for the future (section 5.2) would be to compare different algorithms within the framework presented earlier.

The remainder of this section consists of a description of the principles behind FFNNs. The topology of such a neural network (NN) is discussed first, followed by an explanation of the training and testing procedure. Finally, it is explained how they are employed within this assignment.

3.3.1 Topology

An FFNN consists of multiple layers: an input layer, one or more hidden layers and an output layer. Each of those layers contains one or more neurons. If each neuron in a layer is connected to all neurons in both the previous and the next layer, this layer is called fully-connected. Fig. 3.6 shows an example of an FFNN consisting of fully-connected layers. Note that the term 'feedforward' in FFNN refers to the fact that no recurrent connections exist, such that information can travel in only one direction.

The size of the input and output layer of the NN are determined by the data that is fed into network, $\mathbf{x} = [x_1, x_2, \dots]^T$, and the desired output, $\mathbf{y} = [y_1, y_2, \dots]^T$, respectively. The amount of hidden layers and the number of neurons in each hidden layer

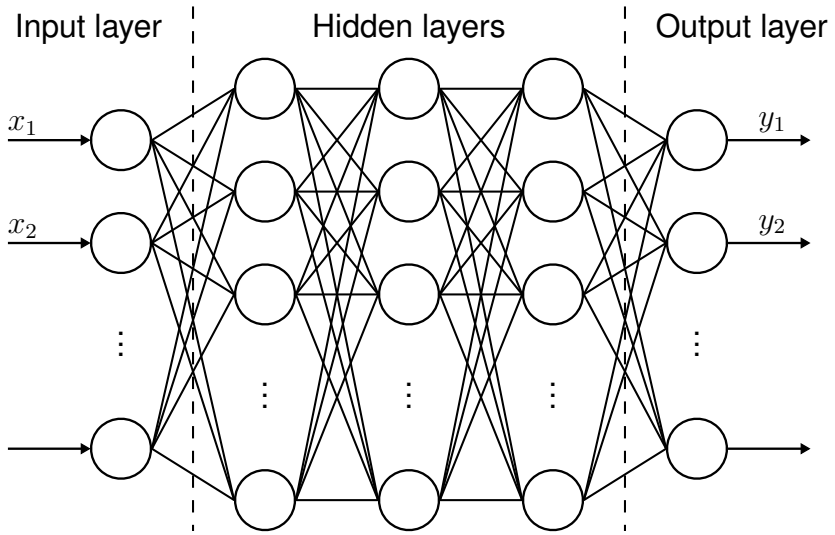


Figure 3.6: Fully connected feedforward neural network.

can be chosen freely. An approach to do this in a structured manner is presented in [12].

Each neuron of the network (except those in the input layer) comprises a sequence of mathematical operations: all elements of the input vector $\mathbf{x}' = [x'_1, x'_2, \dots]^T$ are multiplied with weighting factors $w'_1, w'_2, \dots$. The next step is a summation of all those products and, if desired, a bias term. The output of the summation is the input to a certain activation function. This function can be regarded as some kind of threshold, which produces a certain output y' based on its input. Various common activation functions exist, but they might as well be user defined. A schematic overview of a neuron is shown in Fig. 3.7.

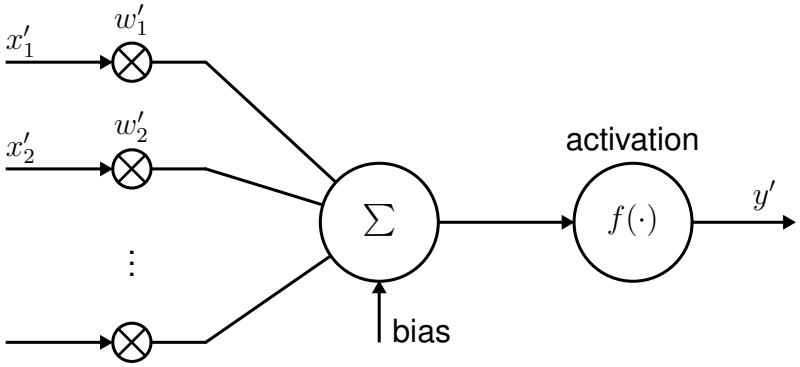


Figure 3.7: Neuron.

It is important to realise that only the weights and the bias, often referred to as the *parameters* of an NN, are adapted during the training stage. All other properties such as the layout of the network, the activation functions, optimizer settings, etc. are to be set before training. These are called the *hyperparameters*.

3.3.2 Supervised learning

This paragraph contains a brief description of how an NN learns from data. As supervised learning is employed within this assignment, only this technique is considered.

Supervised learning is the process of learning a mapping between input and output variables based on input-output pairs, i.e. input data of which the target output is known. The randomly initialized network predicts certain outputs based on the inputs of several input-output pairs. A loss function is used to assess the predictions by comparing them to the true targets. The more accurate the predictions, the lower the loss. An optimizer adjusts the parameters of the network based on the gradient of the loss such that the loss decreases in the next iteration. In order to reduce the computational load, one could use only subset (formally known as mini-batch) of the entire training set in each iteration. If the complete training set has been used once, one epoch has been completed.

In general, the training loss decreases every epoch. However, at some point the network does no longer improve the generic mapping from input to output, but it starts to overfit on the training data. This will degrade the performance of the NN for new observations. A validation set can be used to determine whether this is happening. The data in the validation set is not used during the parameter optimization phase, but it is used to assess the performance of the network afterwards. Based on this assessment, the training process could be terminated. Furthermore, it could be decided to tune the hyperparameters of the network if the performance of the NN does not meet the requirements after training for many epochs. This means, however, that some information of the validation set leaks into the network as well, although implicitly. A third dataset, the test set, is therefore usually employed to get a fair assessment of the performance of the final network. The data in this set is completely new, i.e. none of the observations in this set appear in either the training or validation set. Before performing this final test, the network is usually trained from scratch using both the training and validation data.

3.3.3 Neural networks and the DOA-estimation framework

In this final part of the chapter, it is explained how FFNNs are employed within the framework discussed in section 3.2. The hyperparameters discussed below apply to all networks in the ensemble, unless mentioned otherwise.

Input layer

The input to the NNs in the RAkEL framework is a vector of certain elements of the estimated sensor covariance matrix $\widehat{\mathbf{R}} \in \mathbb{C}^{N\times N}$, similar to e.g. [6]. As the data is created synthetically, this matrix is computed as

$$\begin{aligned}\widehat{\mathbf{R}} &= \frac{1}{T}\sum_{t=1}^{T}\mathbf{y}(t)\mathbf{y}^H(t)\\ &= \frac{1}{T}\,\mathbf{Y}\mathbf{Y}^H\end{aligned} \tag{3.3}$$

with T, $\mathbf{y}(t)$ and $\mathbf{Y}$ according to the data model presented in section 2.1. As $\widehat{\mathbf{R}}$ is a Hermitian matrix, either the upper or lower triangle can be discarded without losing information. In other words, with $r_{i,j}$ being the element at row i and column j for $i,j = 1,\dots,N$ and $\bar{\cdot}$ being the complex conjugate of an entry, it follows that

$$\widehat{\mathbf{R}} = \begin{bmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,N} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N,1} & r_{N,2} & \cdots & r_{N,N} \end{bmatrix}, \tag{3.4}$$

with $r_{i,j} = \overline{r_{j,i}}$. The shaded area in equation 3.4 indicates which elements are used as inputs to the NNs. As only real-valued scalars can be fed into a neuron, each off-diagonal element is associated with 2 neurons in the input layer: one for the real part and one for the imaginary part. In total, N diagonal elements and $(N^2-N)/2$ off-diagonal elements are used, resulting in $N + 2(N^2-N)/2 = N^2$ neurons in the input layer. The input vector $\mathbf{x} \in \mathbb{R}^{N^2}$ is constructed as follows:

$$\mathbf{x} = \begin{bmatrix} r_{1,1} & r_{2,2} & \cdots & r_{N,N} & \Re(r_{1,2}) & \Re(r_{1,3}) & \cdots & \Im(r_{1,2}) & \Im(r_{1,3}) & \cdots \end{bmatrix}^T \tag{3.5}$$

Hidden layers

All hidden layers in the networks are fully-connected. The activation function employed in these layers is the rectified linear unit (ReLU) activation function. This is the most popular activation function in hidden layers of NNs nowadays [12]. The ReLU function $f_{ReLU}(u)$ is defined as

$$f_{ReLU}(u) = max(0,u) \tag{3.6}$$

with u being the output of the summation shown in Fig. 3.7.

The required amount of hidden layers and the number of neurons in those layers depends on the data and/or the performance that is to be achieved, as will be shown in Chapter 4.

Output layer

In section 3.2, it is explained that all classifiers in the ensemble have to deal with a 1-out-of-2^k classification task. This explains why the NNs have 2^k neurons in the output layer: one neuron for each class. The activation function used in the output layer is the softmax funtion. This function is used in many single-label multi-class classification problems. It is defined in such a way that the outputs of all neurons in the output layer add up to 1, such that they can be interpreted as a probability. The predicted class is the one with the highest probability. The output of the i^{th} neuron in the output layer using the softmax activation function $f_{sm,i}(\mathbf{u})$ with $i = 1, \ldots, 2^k$, is defined as

$$f_{sm,i}(\mathbf{u}) = \frac{e^{u_i}}{\sum_{j=1}^{2^k} e^{u_j}}. \tag{3.7}$$

Here, $\mathbf{u} = [u_1, \ldots, u_{2^k}]^T$ is a vector containing the outputs of all summations in the output layer.

Training strategy

Instead of training all networks in the ensemble for a fixed amount of epochs, an early-stopping criterion is employed. If the loss of the validation set, monitored after every epoch, does not decrease anymore, the training stage is terminated. This prevents the networks from overfitting on the training data.

The parameters of the networks are optimized using the Adam optimizer [16] in combination with a *weighted* categorical cross-entropy loss function. Given a vector of target outputs $\mathbf{v} = [v_1, \ldots, v_{2^k}]^T$ and a vector $\hat{\mathbf{v}} = [\hat{v}_1, \ldots, \hat{v}_{2^k}]^T$ containing all probabilities computed by the softmax activation function, the *unweighted* categorical cross-entropy loss D_{CE} is defined as

$$D_{CE}(\mathbf{v}, \hat{\mathbf{v}}) = \sum_i v_i \, log(\hat{v}_i) \tag{3.8}$$

where $i = 1, \ldots, 2^k$. The definition of the softmax function is such that each probability in $\hat{\mathbf{v}}$ is nonzero, which means that the logarithm can always be computed. The target vector contains only zeros except for a single 1 at the entry of the true class. The true class depends on the DOAs of the training observation as well as on the grid segments covered by the considered NN. For example, consider a spatial domain split in $M = 4$ segments: A,B,C and D. Assume one of the networks in the ensemble covers $k = 3$ of those segments, e.g. A,B and D. Furthermore, assume $K = 2$ signals impinge the array, associated with classes A and C according to the procedure discussed in section 3.1. The true class for this observation would be (ABD)=(100), i.e. class 4 (transforming the binary sequence to a decimal number).

The DOA associated with segment C does not affect the target output, as this segment is not covered by the considered network. If multiple DOAs would have been associated with segment A, the target vector would have been the same.

In the weighted version of the loss function, the loss of each of the 2^k classes is scaled by some factor. The reason for this is explained in the next paragraph.

Class imbalance

Before explaining the effect of assigning weights to the loss for each class individually, the variable K_{NN} is introduced. It is defined as the number of segments, out of the k segments considered a by certain NN, in which at least one signal impinges the array. It holds that $0 \leq K_{NN} \leq max(k, K)$. Considering $k = 3$ as an example, it follows that both $K_{NN} = 0$ and $K_{NN} = 3$ each represent 1 class: (000) and (111) respectively. On the other hand, $K_{NN} = 1$ and $K_{NN} = 2$ are both associated with 3 classes, being [(001), (010), (100)] and [(011), (101), (110)]. Since the labelsets are generated randomly, a single observation could correspond to different values of K_{NN} for different networks.

The spatial domain is divided in M segments (Fig. 3.1). In order to accurately estimate the DOAs of K signals, it is required that $K \ll M$. Assuming K signals impinge the array from DOAs $\phi_1, \ldots, \phi_K$ which are i.i.d. random variables of the uniform distribution $U(\phi_{min}, \phi_{max})$, it holds that

$$P(K_{NN} = 0) = \left(\frac{M - k}{M}\right)^K \tag{3.9}$$

with k as defined in section 3.2.2 and $P(K_{NN} = 0)$ being the probability that a certain observation is associated with $K_{NN} = 0$. It can be seen that a large M and a small k and/or K result in a $P(K_{NN} = 0)$ close to 1. This implies that the majority of the observations will be associated with the single class corresponding to $K_{NN} = 0$, i.e. (000). This phenomenon is called class imbalance.

The latter complicates the learning, as minorities are often neglected in classification problems suffering from class imbalance [17]. One way to counteract this problem is to assign weights to the loss for erroneous predictions within the training stage. By making the weights inversely proportional to the support of the classes in the training set, all classes contribute equally much to the loss and the learning algorithm is forced to also focus on the minorities. More advanced techniques for dealing with class imbalance exist, see e.g. [17]. A future investigation could be to determine if the performance of the ML framework could be improved using such techniques.

In chapter 4, the performance of the proposed framework is evaluated and compared to a number of benchmarks using various simulation scenarios.

Chapter 4

Simulations and results

Various simulations were conducted to assess the performance of the ML framework described in chapter 3. The framework is compared against a number of benchmarks by means of performance metrics such as the root-mean-square error (RMSE) and the probability of resolution. Exact definitions of the performance metrics and the benchmarks are presented in appendices A and B respectively.

Section 4.1 contains a description of the simulation settings which apply to all simulations. Then, a scenario is considered in which K is constant over all observations in section 4.2. This constraint is released in the simulations presented in section 4.3.

4.1 General simulation settings

Configuration

The antenna array used within all simulations is an ULA consisting of $N = 8$ isotropic elements. The spacing d between the elements is $\lambda/2$. The array is positioned along the x-axis, according to the definitions presented in Fig. 2.1. The DOAs of the K signals impinging the array are referred to as $\phi_1, \ldots, \phi_K$, with $\phi_{min} < \phi_1 < \cdots < \phi_K < \phi_{max}$. The configuration is visualised in Fig. 4.1.

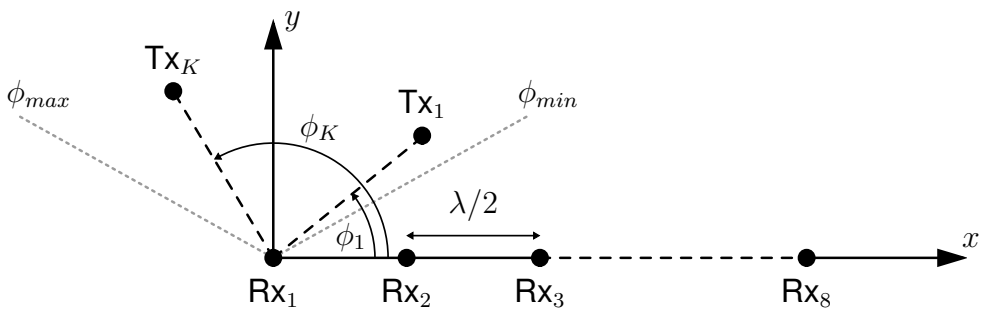


Figure 4.1: K signals impinge an 8-element ULA.

Framework

The considered spatial domain $[\phi_{min}, \phi_{max}]$ is $[30°, 150°]$. It covers 120°, which is common in hexagonal cellular networks. The RAkEL parameter k, i.e. the amount of grid segments covered by each individual NN, is set to $k = 3$. This is suggested in [15], as it is considered to be a good trade-off between performance and computational complexity. The number of layers L (as described in section 3.2.3) equals 5, such that $M < m < 2M$, with M and m as defined in equations 3.1 and 3.2 respectively. The latter is another suggestion in [15], as they observed that increasing m to a value larger than $2M$ does not improve classification performance anymore. The border perturbations are i.i.d. random variables of the uniform distribution $U(-\Delta\phi/4, \Delta\phi/4)$. The minimum width of a segment therefore equals $\Delta\phi/2$.

Data

The data used for training and testing the ensemble of NNs is created using the data model presented in Chapter 2. The estimated sensor covariance matrices used as inputs to the NNs are computed from $T = 100$ snapshots. The DOAs $\phi_1, \ldots, \phi_K$ of the training observations are i.i.d. random variables of the uniform distribution $U(\phi_{min}, \phi_{max})$, unless stated otherwise. The SNR of the training observations is assumed to be a random variable following a discrete log-uniform distribution. In other words, the SNR expressed in dB has $-20, -19, \ldots, 30$ dB as possible values, all with equal probability. The SNR of the test observations is either a log-uniform distributed random variable as well (with a step size of 5 dB), or a fixed value. This varies over the different simulations. If multiple signals impinge the array, they are assumed to have the same power and therefore the same SNR, as already mentioned in section 2.2.

Machine learning

The input layer of all NNs consists of $N^2 = 64$ neurons, whereas $2^k = 8$ neurons are in the output layer. The layout of the hidden layers, determined empirically using a method described in [12], differs over the various simulations. The learning rate of the Adam optimizer is set to 0.001, i.e. the default value as suggested in [16]. The optimization was performed on mini-batches of 32 observations. Of all observations in the training set, 5% is set aside as validation data. The validation data is used for calibrating the threshold of the peak detection algorithm as well. The early stopping criterion employed to counteract overfitting is configured such that training is terminated if the validation loss did not decrease in the last 3 epochs.

4.2 Constant, unknown number of sources

In this section, it is assumed that $K = 2$ for all observations in both the training and the test set. The latter is unknown to the estimator.

The results presented in sections 4.2.1 and 4.2.2 are obtained using a framework with a grid resolution of $\Delta\phi = 2°$. The DOAs for the observations in the test set used in section 4.2.1 are uniformly distributed random variables as well. In section 4.2.2, a different test set is used in order to investigate closely spaced sources in more detail. In section 4.2.3, $\Delta\phi$ is decreased to $0.8°$ to investigate the impact of increasing the resolution of the framework. Finally, section 4.2.4 contains a discussion of how the framework could be adapted to the data if the DOAs are not uniformly distributed.

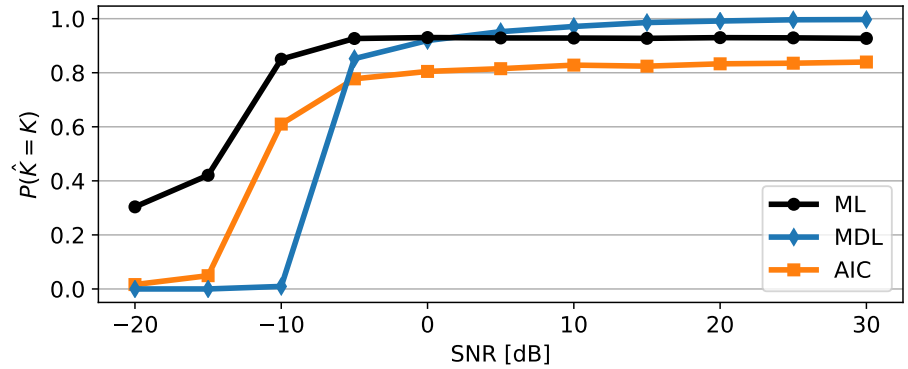
4.2.1 Uniformly distributed random DOAs

The results presented in this section are obtained using a grid resolution of $\Delta\phi = 2°$. Given that $\phi_{max} - \phi_{min} = 120°$, $L = 5$ and $k = 3$, this yields an ensemble of $m = 100$ NNs (equations 3.1 and 3.2). All NNs in the ensemble have 2 hidden layers, one of 64 and one of 36 neurons. Based on empirical research presented in appendix C.1.1, it was observed that training on 10^5 observations is sufficient for this specific scenario. The test set consists of 1.5×10^4 observations for each evaluated SNR.

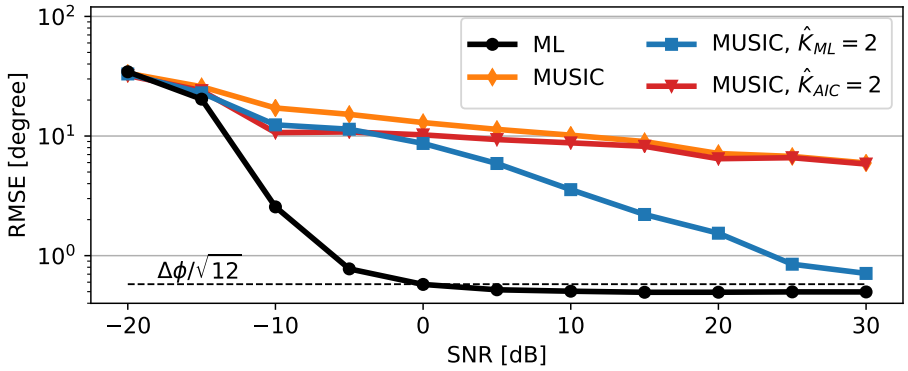
A detailed analysis of the classification performance of the NNs is presented in appendix C.1.2. The analysis confirms that the classifiers suffer from a significant class imbalance, as about 90.3% of the observations are associated with $K_{NN} = 0$. This was expected based on equation 3.9. Furthermore, it is observed that the majority of incorrect predictions for observations associated with $K_{NN} > 0$ implies that a label is not assigned where it should have been, rather than vice versa. This is advantageous, as this merely implies that spectrum peaks are lower than they would have been with perfect classifiers. As the threshold of the peak detection algorithm is adapted to the spectra, the ML framework could, theoretically, compensate for this. Whether it does, is evaluated using a metric relevant in the field of DOA estimation, being the RMSE (appendix A.2). The RMSE is evaluated for both the ML framework and the MUSIC algorithm (appendix B.3). The resolution of the MUSIC algorithm is 0.1°.

The definition of the RMSE is such that it can only be computed if $\hat{K} = K$, i.e. if the estimated number of signals equals the true number of signals. This makes the probability that this is the case, $P(\hat{K} = K)$, a relevant parameter to consider. For the ML framework, $\hat{K}$ is defined as the amount of spectrum peaks higher than the threshold (section 3.2.5), meaning that it is obtained *after* computing the spa-

tial spectrum. However, the MUSIC algorithm (and many other DOA estimation algorithms) requires an estimate of K *before* actually being able to start the computations. The probability $P(\hat{K} = K)$ is therefore evaluated for both the ML framework and two subspace order estimators: the MDL and AIC [4] (appendix B.1). For ease of notation, they will be referred to as, $P(\hat{K}_{MDL} = K)$ and $P(\hat{K}_{AIC} = K)$ respectively, whereas $P(\hat{K}_{ML} = K)$ denotes $P(\hat{K} = K)$ for the ML framework. The RMSE and the $P(\hat{K} = K)$, plotted against the SNR, are shown in Fig. 4.2.



(a) $P(\hat{K} = K)$ vs. SNR



(b) RMSE vs. SNR

Figure 4.2: DOA estimation metrics, $K = 2$, 1.5×10^4 observations per SNR.

As can be observed in Fig. 4.2a, $P(\hat{K}_{ML} = K)$ increases with the SNR of the observations, until it stabilizes at 93% for SNRs of -5 dB and higher. In appendix C.1.3, it is shown that the average spacing between the DOAs in observations for which $\hat{K}_{ML} \neq 2$ equals 2.8° or less (2.4° on average) for SNRs of -5 dB and higher, whereas the average spacing between the DOAs of all observations equals 40° (appendix D.2). It can therefore be concluded that the remaining 7% is dominated by closely spaced sources which cannot be resolved because of the finite resolution of the framework. The ML framework outperforms the AIC estimator for all SNRs, whereas $P(\hat{K}_{MDL} = K) > P(\hat{K}_{ML} = K)$ for SNRs of at least 5 dB. Averaged over all

SNRs, the ML framework correctly estimates the number of signals in 81.8% of the cases, contrary to 69.8% for the MDL. The difference of 12% mainly originates from SNRs smaller than -5 dB. Looking at the RMSE of the framework for these SNRs, (Fig. 4.2b), it can be seen that it approaches 40°. This is the RMSE that would be obtained when estimating the DOAs using sorted uniformly distributed random variables (appendix D.1): $(\phi_{max} - \phi_{min})/\sqrt{3(K+1)} = 120°/\sqrt{3(2+1)} = 40°$. In other words, the DOA estimates at these SNRs hardly contain information.

For SNRs of -5 dB and higher, the RMSE of the DOA estimates of the ML framework is below 1°. For SNRs of 0 dB and higher, it is below $\Delta\phi/\sqrt{12} \approx 0.58°$, which is the RMSE that would be obtained when employing the framework in combination with ideal classifiers without the border perturbations introduced in section 3.2.4 (appendix D.3). This suggests that the border perturbations benefit the DOA estimates, which is confirmed by running another simulation of the same scenario with border perturbations turned off (appendix C.1.4). The difference between the two frameworks in terms of the RMSE is between 0.1° and 0.2° for SNRs of 5 dB and higher, in favour of the framework *with* perturbed borders.

Returning to the RMSE presented in Fig. 4.2b, it can be observed that three graphs are plotted for the MUSIC algorithm. These graphs are obtained by computing the RMSE for

1. all observations, i.e. assuming some perfect K-estimator exists;
2. all observations for which $\hat{K}_{ML} = K = 2$;
3. all observations for which $\hat{K}_{AIC} = K = 2$.

It was decided to use the AIC instead of the MDL as $P(\hat{K}_{AIC} = K) > 0$ for all SNRs. When considering graph 3 from the list above, i.e. the red graph in Fig. 4.2b, it can be seen that the difference between the ML framework and MUSIC, in terms of the RMSE, is at least a factor 10 for all SNRs of -5 dB and higher (in favour of the ML framework). Combining this with Fig. 4.2a, it can be concluded that the ML framework outperforms the combination MUSIC + AIC: neglecting the -20 dB SNR for which the RMSEs of both algorithms approach the RMSE obtained by random guessing, it both holds that $P(\hat{K}_{ML} = K) > P(\hat{K}_{AIC} = K)$ as well as that the RMSE is lower for the ML framework than for MUSIC.

It is important to realise that the observations for which $\hat{K}_{ML} = 2$ and those for which $\hat{K}_{AIC} = 2$ are not the same, as can be concluded from Fig. 4.2a. A more fair comparison between MUSIC and the ML framework is therefore obtained by applying MUSIC to those observations for which $\hat{K}_{ML} = 2$ (graph 2 from the list above, i.e. the blue graph in Fig. 4.2b), such that the graphs are based on the exact same data. The RMSE for the estimates of these observations approaches

the RMSE of the ML framework with an increasing SNR. For an SNR of 30 dB the difference is only 0.2°, still in favour of the ML framework.

After a closer inspection of these results, it was observed that observations with closely spaced sources dominate the RMSEs for the MUSIC algorithm presented in Fig. 4.2b. The latter is explained using a fictitious example, visualised in Fig. 4.3.

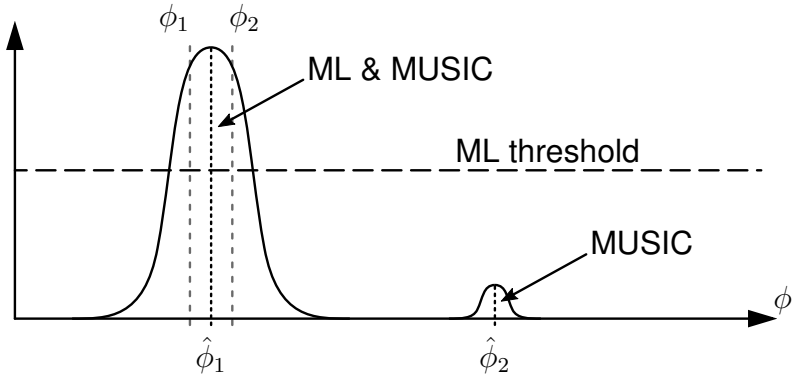


Figure 4.3: Closely spaced DOAs result in a larger RMSE for MUSIC compared to the ML framework.

The DOAs in this example, ϕ_1 and ϕ_2, are closely spaced. Due to the noisy estimate of the sensor covariance matrix employed by both MUSIC and the ML framework and the finite resolution that comes with both methods, the desired peaks in the spatial spectrum merge into a single peak such that the DOAs cannot be resolved. It is assumed that this applies to the spectra computed by both the ML framework and the MUSIC algorithm.

In this situation, the ML framework will return only a single DOA estimate ($\hat{\phi}_1$) as there is only one peak in the spectrum which is higher than the threshold. However, the MUSIC algorithm is told by some other algorithm that $K = 2$ signals impinge the array. It will therefore select the 2 highest peaks of the spectrum and return their arguments as the DOA estimates ($\hat{\phi}_1, \hat{\phi}_2$), even if the second peak is much lower and located at an irrelevant angle. The latter results in large estimation errors dominating the RMSE.

If, for an experiment with closely spaced DOAs, the ML framework successfully resolves the 2 signals whereas MUSIC does not, this will affect the RMSE for MUSIC because of the above. However, when the opposite applies, the experiment is discarded as $\hat{K}_{ML} \neq K$ such that no RMSE can be computed. The latter applies to the 7% of the observations with SNRs of -5 dB and higher for which the average DOA spacing is 2.8° or less.

It can be concluded that the different approaches of MUSIC and the ML framework regarding none-resolvable closely spaced DOAs complicates the comparison of both algorithms. The next paragraph is therefore devoted to closely spaced sources specifically.

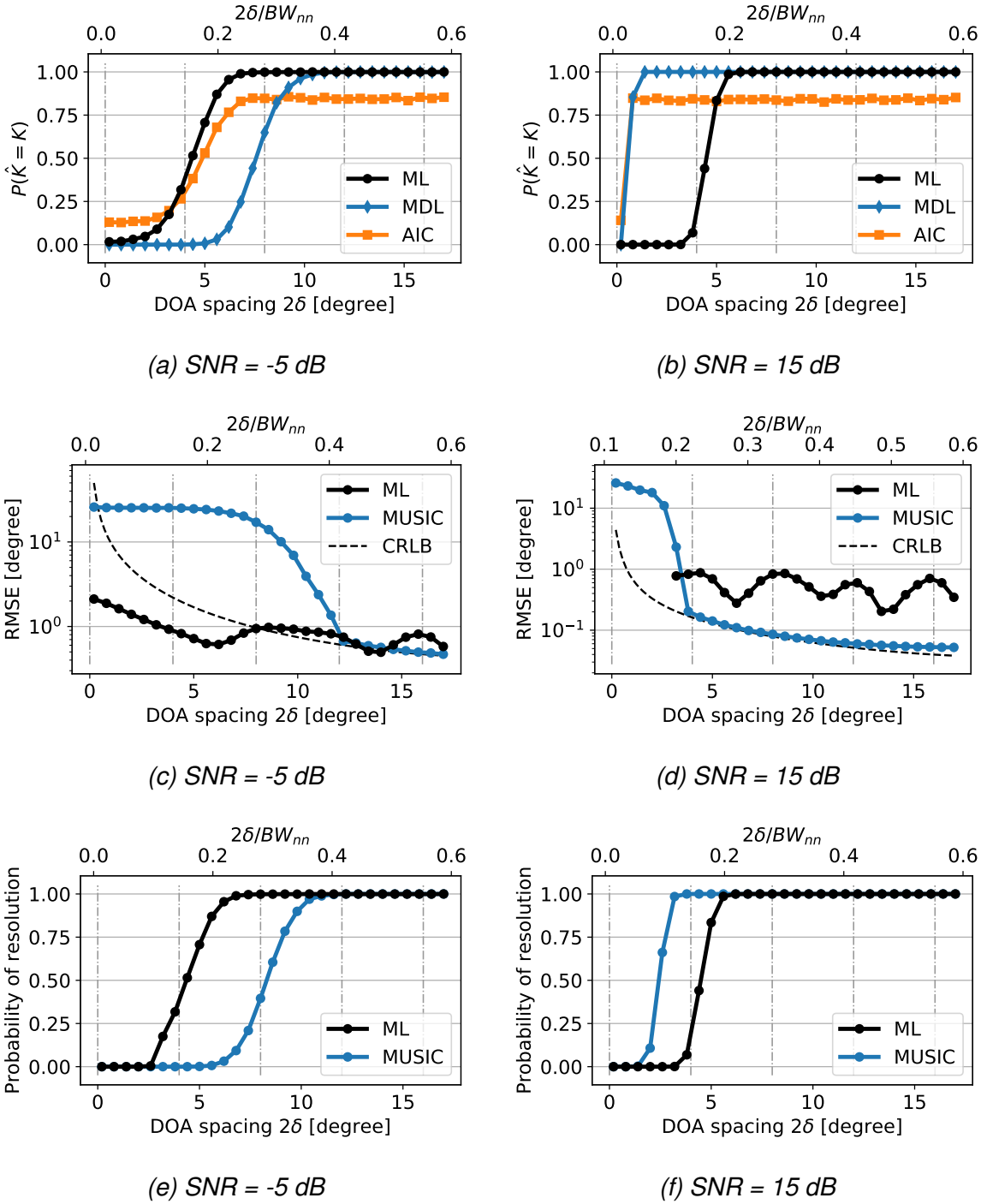
4.2.2 Closely spaced sources

In this paragraph a symmetric scenario is considered in which $\phi_1 = 90 - \delta$ degree and $\phi_2 = 90 + \delta$ degree. The spacing between the two signals, $\phi_2 - \phi_1 = 2\delta$ is increased gradually. Note that this only applies to the test data. The training data is as described in section 4.2.1. Besides the metrics RMSE and $P(\hat{K} = K)$, the probability of resolution is evaluated as well (appendix A.3). The latter metric combines the estimation error and $P(\hat{K} = K)$ into a single number. Two different SNRs are considered, being -5 dB and 15 dB.

The results are presented in Fig. 4.4. Each data point is the average of 5000 observations. Note that the upper x-axes show the DOA spacing relative to the null-to-null beamwidth (BWNN) of the antenna array. Conventional methods such as the beamscan method cannot resolve DOAs with spacings smaller than $BW_{nn}/2$ [18]. For the array employed in this work, $BW_{nn} \approx 28.96°$ (appendix A.3). The vertical grid, shown in all images, represents the spatial grid which is inherent to the framework (neglecting the perturbed borders). In other words, each vertical line indicates that the spacing between the DOAs has increased by $2\Delta\phi = 4°$. The grid resolution is important, as two closely spaced sources can only be resolved if they are separated by at least one grid segment, as explained in section 3.2.4.

In Fig. 4.4a (-5 dB) and Fig. 4.4b (15 dB), $P(\hat{K} = K)$ is shown for both the ML framework as well as for the benchmarks MDL and AIC. Comparing both images, it can be observed that both the MDL and the AIC reach their maximum $P(\hat{K} = K)$ at smaller separations for the higher SNR. To a lesser extent, this also applies to the ML framework: $P(\hat{K} = K) = 1$ for spacings larger than 7.4° (-5 dB) and 5.9° (15 dB). For the 15 dB case, $P(\hat{K}_{ML} = K)$ rises from 0 to 1 as soon as the DOAs are separated by at least 1 grid segment (in this case 2 segments, due to symmetry). The fact that the rise of the edge initiates at slightly smaller separations could be explained by the border perturbations and/or erroneous predictions of the NNs. The slope of the graph of the -5 dB case is more gradual. For this SNR, $P(\hat{K}_{ML} = K) > 0$ for all evaluated separations, even those smaller than $2\Delta\phi$. This does not imply that the ML framework performs better for the -5 dB case though, as it was already reasoned in section 3.2.4 that resolving DOAs with such small separations is theoretically impossible if the classifiers are perfect. The RMSE confirms that false predictions of the classifiers are the cause of the fact that $P(\hat{K}_{ML} = K) > 0$ for separations smaller than $2\Delta\phi$.

The RMSEs for both considered SNRs are shown in Fig. 4.4c and Fig. 4.4d. The ML framework is compared to MUSIC and the Cramér-Rao lower bound (CRLB), which is the minimum variance for unbiased estimators (in fact, the square root of the CRLB is shown). The MUSIC algorithm was applied to all observations in the test set, i.e. assuming some perfect K estimator exists. Looking at RMSE



(a) SNR = -5 dB

(b) SNR = 15 dB

(c) SNR = -5 dB

(d) SNR = 15 dB

(e) SNR = -5 dB

(f) SNR = 15 dB

Figure 4.4: ML framework performance vs. signal separation for SNRs of -5 and 15 dB. (a),(b), $P(\hat{K} = K)$; (c),(d), RMSE; (e),(f) probability of resolution.

of the ML framework for the -5 dB case, it can be observed that it is bigger than 1° for separations smaller than $2\Delta\phi$ (i.e. left of the second vertical grid line). As the resolution of the framework $\Delta\phi = 2°$, the maximum RMSE would be 1° for ideal classifiers. It can therefore be concluded that $P(\hat{K} = K) > 0$ for these small separations results from erroneous predictions of the NNs. Despite these errors, the ML framework clearly outperforms MUSIC for separations smaller than $0.4BW_{nn}$. For larger separations, the framework seems to be limited by its grid resolution of 2°. The fact that the RMSE of the ML framework is lower than the CRLB indicates that the framework is biased. This is inherent to the discrete grid on which the framework is based, as DOA estimates will either be the centre of a grid segment, or some angle close to the border between two classes.

For the 15 dB case, MUSIC outperforms the ML framework (in terms of the RMSE) for nearly all separations. The impact of the finite resolution of the framework can be observed, as the peaks and valleys in its graph align with the segment borders and segment centres respectively. However, the RMSE is below 1° for all peaks, confirming once more that the border perturbations benefit the DOA estimates. A maximum improvement of 0.4° is observed (0.6° instead of 1° at the 4th vertical grid line). Finally it can be seen that the RMSE for MUSIC cannot follow the CRLB for separations larger than approximately $0.3BW_{nn}$. This can be explained by the 0.1° resolution of the scanangles at which the MUSIC spectra are evaluated.

The probability of resolution is depicted in Fig. 4.4e and Fig. 4.4f. For the -5 dB scenario, the ML framework achieves 100% at DOA spacings of at least 7.4°, contrary to 11.6° for the MUSIC algorithm. For separations smaller than $0.1BW_{nn}$, the DOAs are never resolved. However, $P(\hat{K}_{ML} = K) > 0$ for these spacings, implying that the estimates are not accurate enough. The reader is referred to the definition of the probability of resolution (appendix A.3) for a formal definition of 'accurate'. For the 15 dB case, the ML framework resolves both signals in all observations if the spacing is at least 5.9°, contrary to 3.5° for MUSIC. Note that the probability of resolution is identical to $P(\hat{K}_{ML} = K)$ (Fig. 4.4b) for this SNR. In other words, if the source number is estimated correctly, the DOA estimates are accurate as well. Note that the results shown for MUSIC assume that a perfect K-estimator exists. In practice, MUSIC can only be applied in combination with, e.g. a subspace order estimator, meaning that the probability of resolution could be lower than shown here.

It can be concluded that the ML framework outperforms the MUSIC algorithm regarding closely spaced sources for an SNR of -5 dB, whereas the opposite holds for 15 dB. For the latter SNR, the resolution of the framework seems to be a limiting factor, as both $P(\hat{K}_{ML} = K)$ and the probability of resolution rise from 0 to 1 as soon as the spacing exceeds $2\Delta\phi$. Whether increasing the resolution could improve the performance of the ML framework is presented in the next section.

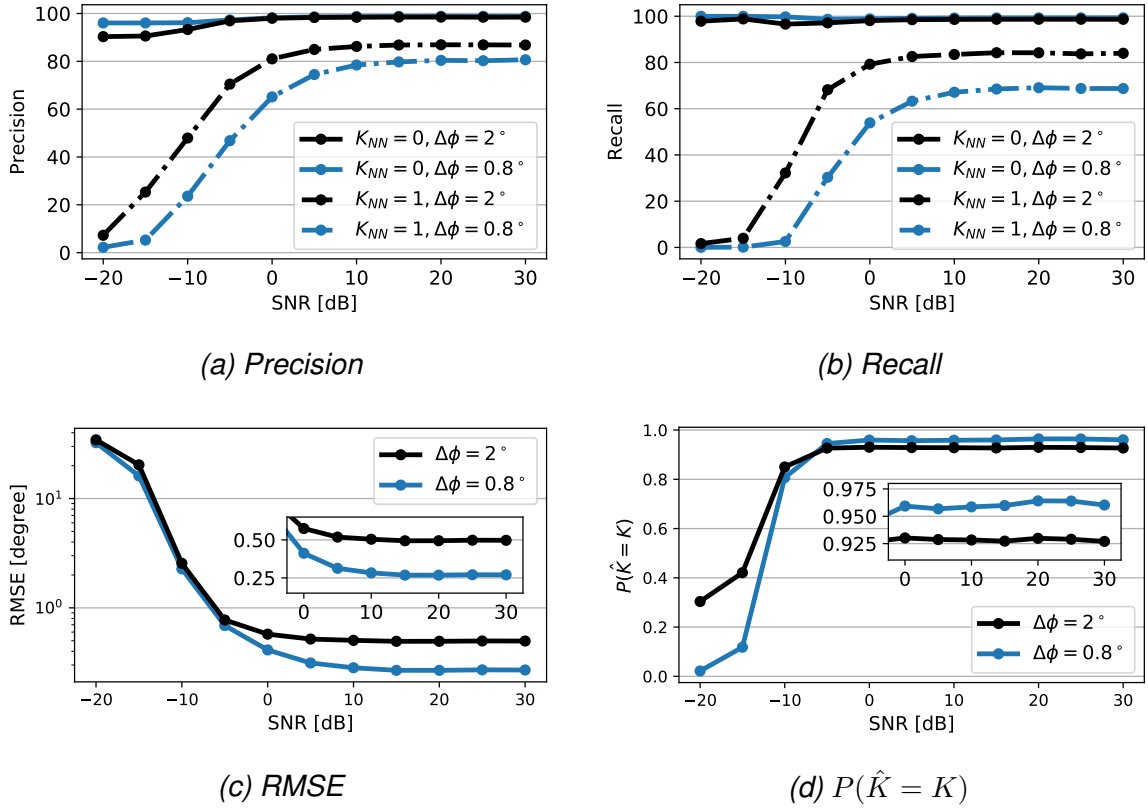
4.2.3 Increasing the grid resolution

Based on the results presented in sections 4.2.1 and 4.2.2, it was decided to investigate if increasing the resolution of the framework, i.e. decreasing $\Delta\phi$, could increase its performance. Decreasing $\Delta\phi$ implies an increased amount of grid segments M (equation 3.1). Considering uniformly distributed random DOAs with $K = 2$ (as in section 4.2.1), it follows that $P(K_{NN} = 0)$ increases as well (equation 3.9). In other words, increasing the resolution of the framework comes with an increase of the class imbalance. Less observations will therefore be associated with the minority classes, i.e. classes of the clusters $K_{NN} = 1$ and $K_{NN} = 2$, if the size of the training set is kept the same. Learning an accurate mapping for these classes therefore becomes more and more complex with smaller values of $\Delta\phi$.

Despite an increase of M, the task of each classifier (in this work NN) in the ensemble remains the same: picking the correct class out of 2^k possible classes. In other words, if the networks in a framework with increased resolution achieve similiar behaviour as the networks in the 2° resolution framework, the RMSE would automatically decrease. Furthermore, it would imply that closely spaced DOAs could be resolved for smaller spacings, i.e. that the graph representing the ML framework in Fig. 4.4f shifts to the left. Assuming the latter holds, it can be observed that a grid resolution of $\Delta\phi = 0.8^\circ$ would be sufficient to outperform the MUSIC algorithm in terms of the probability of resolution at an SNR of 15 dB.

In Fig. 4.5, two frameworks are compared: the framework with a resolution of $\Delta\phi = 2^\circ$ employed in sections 4.2.1 and 4.2.2, and a 'high resolution' framework with $\Delta\phi = 0.8^\circ$. Based on empirical research, it was decided to increase the training set for the latter framework to 4×10^5 observations. Furthermore, the network layout is increased to 3 hidden layers, consisting of 84, 84 and 36 neurons.

Besides the conventional DOA estimation metrics RMSE and $P(\hat{K} = K)$, the performance of both frameworks is evaluated by means of the metrics precision and recall as well (appendix A.1). These are measures for the classifiers' exactness and completeness respectively. In other words, they are an indication of the quality of the predictions performed by the NNs, before they are converted to DOA estimates. By definition, both metrics are computed for each of the 2^k classes individually. Classes within a cluster of a certain K_{NN} share that one or more signals impinge the array from DOAs associated with K_{NN} of the k segments. However, the position of these segments in the spatial domain is random. It was therefore decided to average the metrics for all classes within a cluster by means of a macro average (appendix A.1). As these metrics are computed for each network individually, another average is applied such that a single number is obtained for a certain K_{NN} at a certain SNR. Both metrics are only shown for the clusters $K_{NN} = 0$ and $K_{NN} = 1$, as 99.8% of the observations corresponds to one of those clusters if $\Delta\phi = 2^\circ$ and even more for



(a) Precision *(b) Recall*

(c) RMSE *(d)* $P(\hat{K} = K)$

Figure 4.5: Performance metrics for frameworks of $\Delta\phi = 2^\circ$ and $\Delta\phi = 0.8^\circ$.

the higher resolution (appendix D.4). The precision and recall are shown in Fig. 4.5a and Fig. 4.5b respectively.

It can be observed that both the precision and the recall for observations corresponding to $K_{NN} = 0$ are at least 97% for SNRs of -5 dB and higher, for both resolutions. A high performance was to be expected, as classifiers are biased towards the majority class in problems with a significant class imbalance [17]. It implies that the framework is good at estimating at which angles signals *do not impinge* the array. However, the classes of the cluster $K_{NN} = 1$ are most important to decide from which angles signals *do impinge* the array, as correct predictions for observations corresponding to these classes will result in peaks in the spectra at the correct angles. For classes of this cluster, the precision of the 0.8° resolution framework is above 78.4% for SNRs of 10 dB or higher, with a maximum of 80.7% for an SNR of 30 dB. The NNs in the low resolution framework achieved a precision above 86.2% for SNRs of at least 10 dB. A similar trend is observed for the recall, although the differences are bigger. For SNRs of at least 5 dB, the recall was above 82.5% for the low resolution framework, whereas it is between 63.2% and 68.7% after increasing the resolution.

To determine if the decreased precision and recall have an impact on the quality of the DOA estimates, the RMSE and $P(\hat{K} = K)$ are evaluated. In Fig. 4.5c, it

can be observed that increasing the resolution has reduced the RMSE, with a difference of about 0.25° for SNRs of 10 dB and higher. However, the minimum RMSE achieved (0.25°) is not below the RMSE for perfect classifiers without perturbations, $\Delta\phi/\sqrt{12} = 0.8°/\sqrt{12} \approx 0.23°$. The low resolution framework did achieve this limit (0.50° vs. 0.58°, appendix D.3). The probability that $\hat{K} = 2$ (Fig. 4.5d) improved as well, as it increased by about 3% for SNRs of 0 dB and larger. The fact that it decreased for the lowest SNRs does not matter, as the RMSE indicates that the DOA estimates at these SNRs do not contain information anyway. Note that for SNRs of 0 dB and higher, $\hat{K} = 2$ for more than 95% of the observations with and RMSE below 0.5°, while for the same SNRs, the precision is between 65.1% and 80.7% and the recall between 53.9% and 68.7%. Since each grid segment is covered by $L = 5$ classifiers and the threshold of the peak detection algorithm is adapted to the data, the framework successfully compensates for the relatively low performance of the classifiers.

The origin of the increase in $P(\hat{K} = K)$ can be explained by investigating the probability of resolution for closely spaced sources once more. It is evaluated for the frameworks of both resolutions, as well as for the MUSIC algorithm. The results are shown in Fig. 4.6.

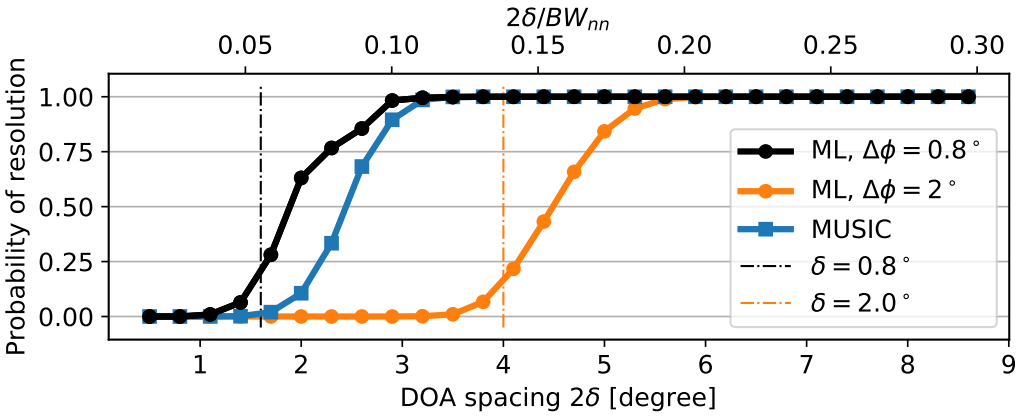


Figure 4.6: Probability of resolution for various $\Delta\phi$, SNR = 15 dB.

The two vertical grid lines represent a DOA spacing of $2\Delta\phi$. It can be observed that, similar to the low resolution framework, the probability of resolution for the high resolution framework rises from 0 to 1 as soon as the DOA separation has increased beyond $2\Delta\phi$. The high resolution framework achieves a 100% probability of resolution for spacings larger than 3.2°, contrary to 5.9° for the low resolution framework. The high resolution framework outperforms the MUSIC algorithm as well (100% probability of resolution for spacing larger than 3.5°), whereas this only applied to the lower SNR (-5 dB) for the 2° framework (section 4.2.2). The biggest

difference between MUSIC and the high resolution ML framework is obtained at a DOA spacing of 2°, where the ML framework achieves a probability of resolution of 63%, contrary to 11% for MUSIC.

It can be concluded that the performance of the ML framework can be increased by increasing its resolution, although this is at the expense of more training data being required, at least for the data, learning algorithm, training strategy, etc. considered here. A recommendation for the future would be to investigate if the need for extra training data could be diminished by advanced class imbalance techniques such as oversampling or synthetic data generation [17]. Furthermore, it would be useful to find a relation between the performance of the classifiers, e.g. in terms of the classification metrics presented in appendix A.1, and the DOA estimation performance metrics RMSE and $P(\hat{K} = K)$. The results presented in this section (Fig. 4.5 specifically) indicate correlation between the two, but further research is required to determine the impact of, e.g., the number of layers L in the framework. If such a relation is known, one could use the validation set to determine (already during training stage) if satisfactory DOA estimation performance will be achieved for the considered settings (grid resolution, training strategy, etc).

4.2.4 Laplace distributed random DOAs

In this final paragraph of the current section, the DOAs of the signals are no longer assumed to be uniformly distributed over the spatial domain. Instead, it is assumed that they are Laplacian distributed, as this is the most common model for the angular dispersion at a base station [19]. In [20], it is stated that the root-mean-square (rms) angular spread σ for line-of-sight (LOS) situations in a microcell is typically between 5° and 20°. With the distribution being centred around the mean μ, the pdf $f(x)$ of the Laplace distribution is defined as

$$f(x) = \frac{1}{\sqrt{2}\sigma} \exp\left(- \frac{\sqrt{2}|x - \mu|}{\sigma} \right) \tag{4.1}$$

with support $x \in \mathbb{R}$. Its cdf $F(x)$ is defined as

$$F(x) = \begin{cases} \frac{1}{2} \exp\left(\frac{\sqrt{2}(x-\mu)}{\sigma} \right) & \text{if } x \leq \mu \\ 1 - \frac{1}{2} \exp\left(- \frac{\sqrt{2}(x-\mu)}{\sigma} \right) & \text{if } x > \mu \end{cases} \tag{4.2}$$

The cdf has the same support as the pdf, i.e. all real numbers. However, in the given application, only values in the domain $[\phi_{min}, \phi_{max}]$ will be used. The pdf and cdf are

therefore to be truncated, which is done according to

$$f_{tr}(x) = \frac{f(x)}{F(\phi_{max}) - F(\phi_{min})} \tag{4.3a}$$

$$F_{tr}(x) = \frac{F(x) - F(\phi_{min})}{F(\phi_{max}) - F(\phi_{min})} \tag{4.3b}$$

with $f_{tr}(x)$ and $F_{tr}(x)$ being the truncated pdf and cdf respectively.

The Laplace distribution implies that signals are more likely to impinge the array closer to the mean $\mu = (\phi_{min} + \phi_{max})/2$ of the distribution. For example, for an rms angular spread of $\sigma = 20°$, with $\phi_{min} = 30°$, $\phi_{max} = 150°$ and $\mu = 90°$, it can be computed using equations 4.1-4.3 that 50% of the DOAs are within the inner 19.2° of the spatial domain, and 90% is within the inner 61.7°. For smaller rms angular spreads, this effect is even stronger. If multiple signals impinge the array, they will be closer to each other on average. For the same angular domain, the average spacing between 2 signals equals 19.7° for the Laplace distribution ($\sigma = 20°$), contrary to 40° for the uniform distribution.

It is suggested to adapt the framework to the distribution of the DOAs. By defining an equally spaced grid of M pieces between 0 and 1, and relating each point of this grid to a certain angle within the spatial domain via the inverse truncated cdf, an irregular spatial segmentation is obtained. This procedure is visualised in Fig. 4.7. The irregular segmentation has a number of advantages compared to a uniform segmentation. First of all, the class imbalance is minimized, as the probability that the DOA of a signal is associated to a certain segment is equal for all segments, i.e. $1/M$. Furthermore, for a certain number of segments M, the width of the segments will be smaller (compared to a uniformly segmented framework using the same M) at angles where most of the signals impinge the array. This could, at least theoretically, result in more accurate DOA estimates for the majority of the signals. In addition, it could increase the probability of resolution for closely spaced sources, which will appear more frequent.

For example, the framework employed in sections 4.2.1 and 4.2.2 had a resolution of $\Delta\phi = 2°$, which results in $M = 120°/2° = 60$ segments. Consider another framework of 60 segments, adapted to Laplace distributed data with a 20° rms angular spread. It can be computed that for angles $69.4° < \phi < 110.6°$, the segments will have a width less than $2°$. As a result, the DOAs could could theoretically be estimated with smaller estimation errors for 78% of the observations if a single signal impinges the array. The opposite applies to the remaining 22%. Note that latter percentage decreases if the angular spread of the Laplace distribution decreases.

The latter example was simulated to verify the idea presented above. Two signals with Laplacian distributed random DOAs ($\sigma = 20°$,$\mu = 90°$) are assumed to impinge the array. The framework is constructed with $M = 60$, such that it can be compared

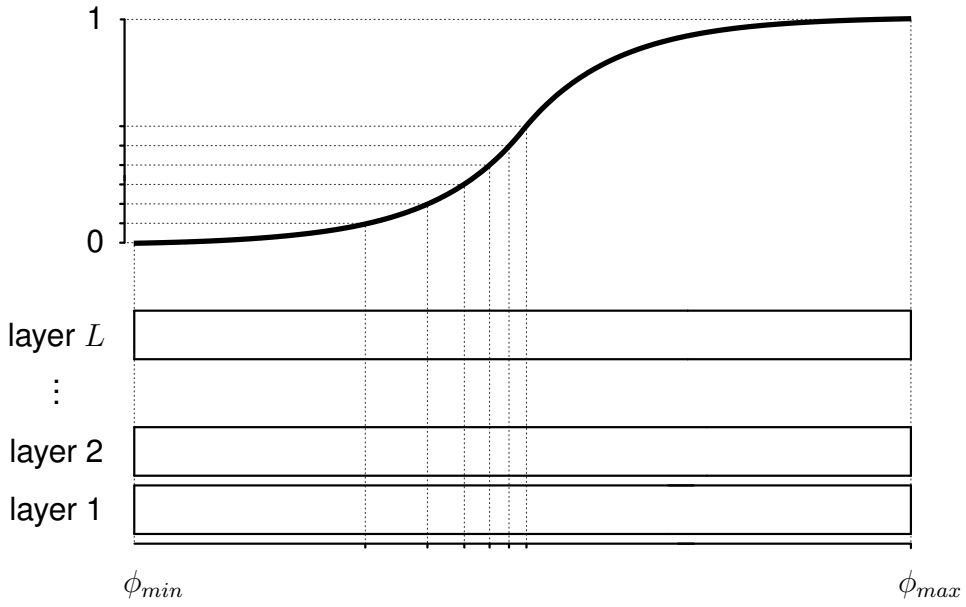


Figure 4.7: Spatial segmentation based on the CDF of the Laplace distribution.

to the framework presented in sections 4.2.1 and 4.2.2. For both frameworks, it holds that $k = 3$ and $L = 5$, which implies that the number of networks equals $m = 100$. The NNs in the Laplace-based framework consist of 3 hidden layers (84,84,36 neurons). The probability of resolution for both frameworks, as well as for the MUSIC algorithm, is presented in Fig. 4.8. Again, the vertical grid lines indicate where the spacing has increased beyond 2 grid segments.

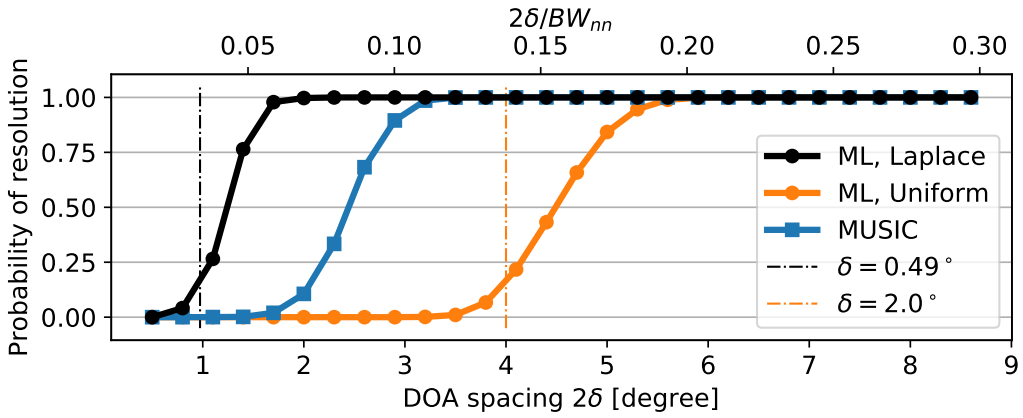


Figure 4.8: Probability of resolution for ML frameworks with $M = 60$ segments, adapted to the DOA distributions, SNR = 15 dB.

It can be observed that the Laplace based framework outperforms both the MUSIC algorithm and the uniformly segmented framework in terms of its capability to resolve closely spaced sources. The Laplace based framework resolves both signals successfully for all observations with spacings of $0.09BW_{nn} \approx 2.6°$ and larger.

MUSIC achieves a 100% resolution probability at $0.13BW_{nn} \approx 3.8°$. The uniform framework with a resolution of $\Delta\phi = 2°$ resolves all signals as soon as the spacing is larger than $5.9°$ (Fig. 4.6).

A number of important remarks are to be made. Firstly, the Laplace-based framework was trained on 2×10^5 observations, contrary to 1×10^5 for the uniform framework. It could be that a uniform framework with a smaller $\Delta\phi$ could be employed when increasing the training set to 2×10^5 observations. However, the $0.8°$ resolution framework presented in section 4.2.3, trained on 4×10^5 observations, does not outperform the Laplace based framework in terms of the probability of resolution for closely spaced sources either. Furthermore, the increased performance near the centre of the domain is at the expense of the estimation accuracy for DOAs close to its edges. Segment 1 and 60, according to the definition in Fig. 3.1, have a width of 16.8° and will therefore result in large estimation errors. In practice, a certain required maximum error (in degrees) for a certain part of the spatial domain, together with the rms angular spread, will determine the required number of grid segments M. Whether the framework actually achieves satisfactory performance for this required M depends on the training strategy and the amount of available data, where the need for additional data might be diminished by improving the strategy.

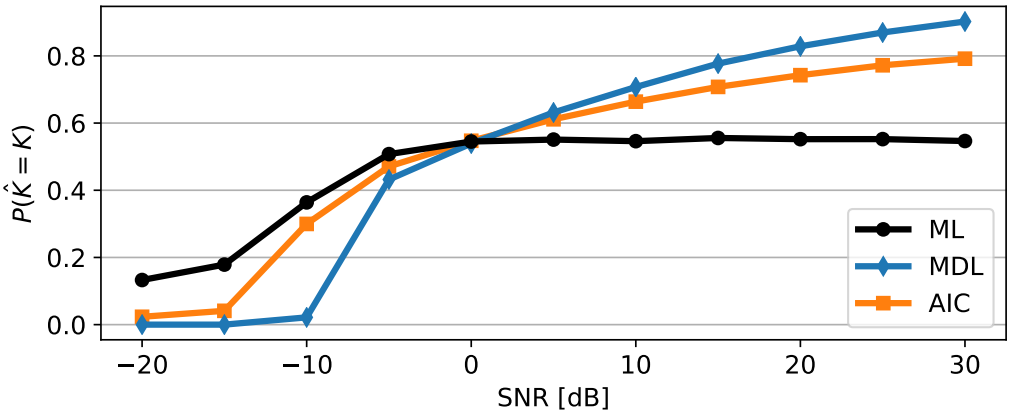
The results presented in Fig. 4.8 indicate that adapting the framework to the data has a significant effect on its performance for the given dataset. The MUSIC algorithm does not have this freedom, such that its DOA estimates for a certain observation are independent of all other observations. For the current training strategy (section 3.3.3), the amount of required training data seems to increase with an increased class imbalance (section 4.2.3). Adapting the segmentation to the distribution of the DOAs is therefore recommended if a limited amount of data is available. Whether this also benefits the DOA estimates depends on the requirements of a certain application.

4.3 Variable, unknown number of sources

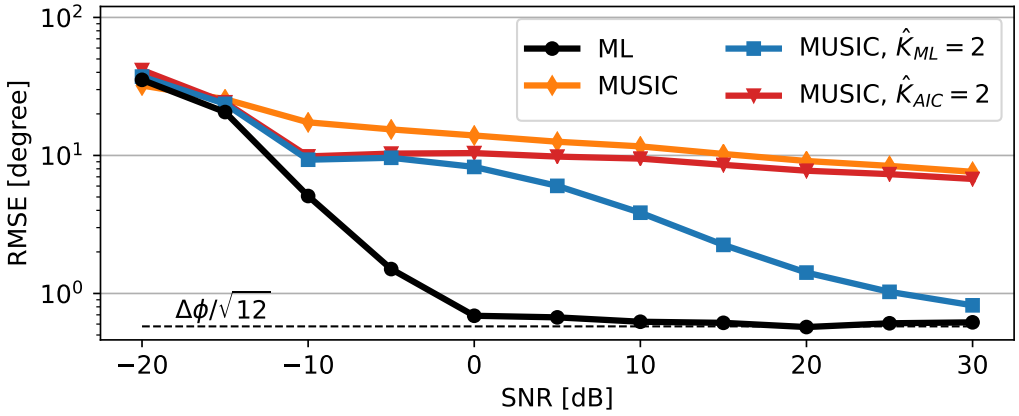
The number of signals K impinging the array was constant for all observations in section 4.2. In this final section, this constraint is released. Instead, $K \sim U(1, N-1)$, i.e. $K = 1, \ldots, N-1$. The antenna array is the same as in the previous section, such that $N = 8$. The DOAs of the signals are again i.i.d. random variables of the uniform distribution $U(\phi_{min}, \phi_{max})$. This applies to observations in both the train and test set. The networks in the ensemble are trained on 10^6 observations. They consist of 4 hidden layers of 84, 84, 84 and 50 neurons respectively. As before, all signals within an observation have the same SNR. The SNR is a random variable as well, with the same distribution as in section 4.2.

4.3.1 Uniformly distributed random DOAs

Similar to the 2 signal scenario presented in section 4.2, the RMSE is evaluated in combination with $P(\hat{K} = K)$. The test set has been increased to 5×10^4 observations for each SNR. The results are visualised in Fig. 4.9.



(a) $P(\hat{K} = K)$ *vs. SNR*



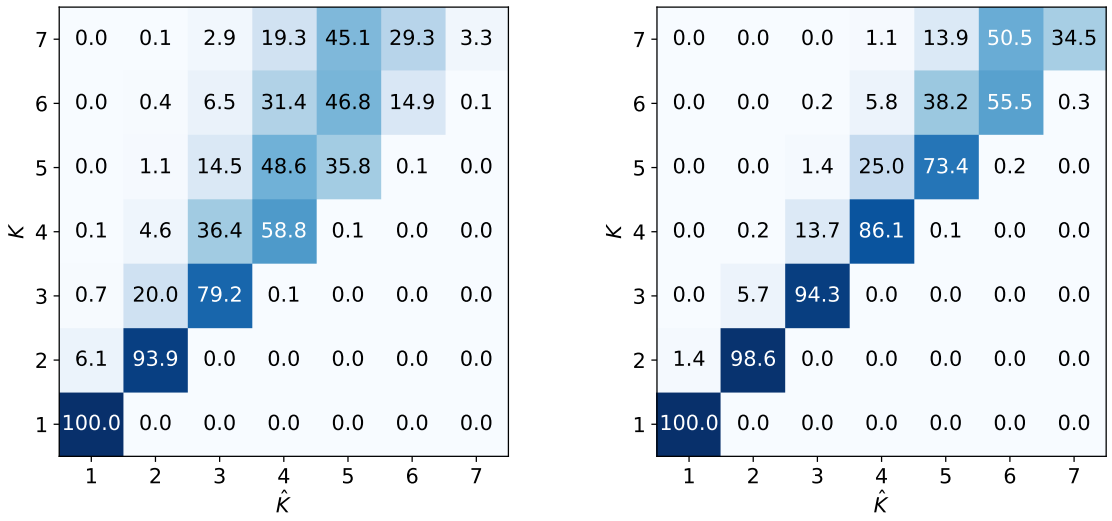
(b) RMSE vs. SNR

Figure 4.9: DOA estimation metrics, $K \sim U(1, 7)$, 5×10^4 observations per SNR.

The RMSE presented in Fig. 4.9b shows many similarities with the Fig. 4.2b: again the RMSE for the ML framework is at least a factor 10 smaller for SNRs of 0 dB and larger, compared to the MUSIC algorithm employed in combination with the AIC or some ideal K-estimator. Furthermore, MUSIC approaches the ML framework when applying it only to observations for which $\hat{K}_{ML} = K$. An explanation of this behaviour was presented in section 4.2.1. Contrary to the 2-signal case, the RMSE of the ML framework does not get below $\Delta\phi/\sqrt{12} \approx 0.58°$, although the differences are within 0.2° for SNRs of 0 dB and larger. The latter will be analysed in more detail later in this section.

Looking at the plot of $P(\hat{K} = K)$ vs. SNR (Fig. 4.9a), it can be seen that this metric increases with the SNR, until it stabilizes at 55% for SNRs of 0 dB and higher. A similar trend was observed in the 2 signal case, although the framework achieved a $P(\hat{K} = K)$ of 93% for SNRs of -5 dB and higher. The reader is reminded of the fact that the observations for which $\hat{K} \neq 2$ were observations with closely spaced sources, i.e. 2.4° on average for an SNR of -5 dB and higher. With an increased amount of signals impinging the array, the average spacing between them decreases (appendix D.2). The latter complicates the estimation as the resolution of the framework, $\Delta\phi = 2°$, is finite. A lower $P(\hat{K}_{ML} = K)$ for this scenario, relative to the 2-signal scenario (Fig. 4.2a), was therefore to be expected. The subspace order estimators compute $\hat{K}$ from the eigenvalues of estimated sensor covariance matrix and therefore they do not suffer from a finite resolution. As a result, both algorithms outperform the ML framework for SNRs higher than 0 dB. In fact, the MDL achieves 90% accuracy for an SNR of 30 dB.

In the remainder of this section, the relation between the performance of the ML framework and the number of signals K is investigated. This is done using a test set of 1.5×10^5 observations of a 15 dB SNR. At this specific SNR, $P(\hat{K} = K)$ equals 55.6% for the ML framework, contrary to 70.8% for the AIC and 77.7% for the MDL (Fig. 4.9a). To investigate if this is indeed solely caused by the finite resolution of the framework, the distribution of $\hat{K}$ for different values of K is evaluated. This is done by means of a heatmap, which is computed for both the ML framework as well as for the MDL. They are shown in Fig. 4.10. The heatmaps are normalized row-wise, such that each row adds up to 100%.



(a) ML framework *(b) MDL*

Figure 4.10: K vs. $\hat{K}$, SNR = 15 dB.

The diagonal of the heatmaps can be interpreted as the metric $P(\hat{K} = K) \times 100\%$. This number decreases with an increasing value of K for both methods. Both achieve a 100% accuracy when only one signal impinges the array. However, the ML framework correctly estimates the number of signals in only 3.3% of the cases if $K = 7$, contrary to 34.5% for the MDL. Furthermore, it can be observed that if $K = 2$, this is correctly estimated by the ML framework in 93.3% of the observations. In the 2 signal scenario, this was the case in only 92.8% of the observations (Fig. 4.2a). It is expected that this is caused by the increased training set (10^6 instead of 10^5 observations) in combination with the increased network layout (4 instead of 2 hidden layers).

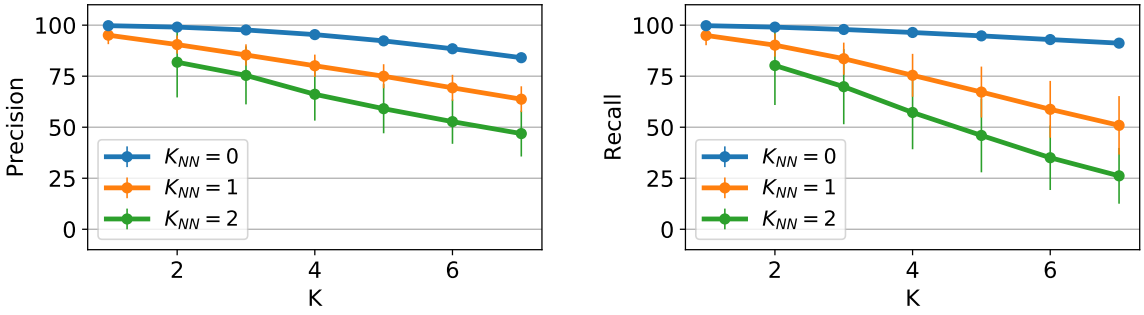
It was already mentioned that the average spacing between neighbouring DOAs decreases if K increases. A certain spacing between all DOAs is required (also theoretically) because of the finite resolution of the framework. The probability that the smallest spacing (denoted as ψ_{min} to prevent confusion with ϕ_{min}) between neighbouring uniformly distributed random DOAs is bigger than this required spacing ψ can be computed as

$$P(\psi_{min} \geq \psi) = \left(1 - \frac{(K-1)\psi}{(\phi_{max} - \phi_{min})}\right)^K. \tag{4.4}$$

In section 4.2.2 it was shown that if $K = 2$ and $\Delta\phi = 2°$, $P(\hat{K} = K) = 1$ for spacings larger than $0.21BW_{nn} \approx 5.9°$ (Fig. 4.4b). Assuming that this also applies if $K = 7$, it would hold that $P(\psi_{min} \geq 5.9°) \times 100\% \approx 8\%$. In other words, even if the ensemble of NNs would achieve the same performance (in terms of $P(\hat{K} = K)$ versus DOA spacing) as the ensemble discussed in the previous section despite the added complexity (i.e. a varying K), $\hat{K}$ would be correct in only 8% of the observations with $K = 7$, just because of the construction of the framework itself.

However, instead of 8%, $\hat{K}$ equals K in only 3.3% of the cases. Furthermore, it was already observed in Fig. 4.9b that the lower bound for perfect classifiers without border perturbations $\Delta\phi/\sqrt{12}$ was not achieved for any SNR, contrary to the 2-signal scenario. Both phenomena could be explained by erroneous predictions of the classifier ensemble. To investigate if this is indeed the case and how this relates to the number of signals K, the precision and recall are evaluated again. Both metrics are plotted against K in Fig. 4.11. The confusion matrix from which these metrics are derived is shown in appendix C.2.1. This matrix shows that only 38 of the 1.5×10^5 observations, averaged over all NNs in the ensemble, are associated to $K_{NN} = 3$. It was therefore decided to only show the metrics for $K_{NN} = 0, \ldots, 2$.

It was already shown for the 2 signal scenario that the support of training (and test) observations associated with a certain K_{NN} decreases with an increasing K_{NN} (appendix C.1.2). This also applies to the simulation considered here, and it is therefore expected that class imbalance is again the cause of the decreasing precision



(a) Precision vs. SNR *(b) Recall vs. SNR*

Figure 4.11: Classifier performance vs. K for various K_{NN}, SNR = 15 dB.

and recall with increasing K_{NN}, Fig. 4.11a and Fig. 4.11b respectively. However, the fact that both metrics decrease with K as well cannot be related to an imbalanced amount of observations being associated with each K, as K is a uniformly distributed random variable. Below, a qualitative reasoning is presented which explains the relation between the precision/recall and the DOA estimates. Further research is required to determine the minimum required precision and recall for certain DOA estimation performance, and their relation to e.g. the number of layers in the framework.

Aided by the confusion matrix presented in appendix C.2.1, it can be concluded that a recall below 100% for $K_{NN} = 1$ and $K_{NN} = 2$ merely implies a decreased height of the spectrum peaks. This is, to some extent, accounted for by the fact that the threshold of the peak detection algorithm is adapted to the data. However, a recall below 100% for observations associated with $K_{NN} = 0$ implies that spectrum peaks appear at unwanted angles, i.e. angles more than $\Delta\phi/2$ away from the true DOAs. For the precision, the opposite reasoning applies: a precision lower than 100% for $K_{NN} = 0$ could be compensated for to some extent, whereas a low precision for $K_{NN} = 1$ and $K_{NN} = 2$ results in unwanted peaks. As long as the unwanted peaks are below the threshold and the wanted peaks are above, this is not a problem. However, as both the recall for $K_{NN} = 0$ as well as the precision for $K_{NN} = 1$ and $K_{NN} = 2$ decrease with an increasing K, the probability that such an unwanted peak gets above the threshold increases with K. In other words, a high $P(\hat{K} = K)$ does not necessarily imply a good DOA estimator, as it could be that peaks at correct angles are discarded whereas peaks at unwanted angles are not. Whether the latter applies to the simulation considered here, could be investigated by means of the RMSE: if for a certain observation $\hat{K} = K$, but (part of) the $\hat{K}$ DOA estimates originate from spectrum peaks at angles which are not within $\Delta\phi/2$ of the true DOAs, this results in relatively large estimation errors. As a result, the RMSE will increase. In Fig. 4.12, it can be seen that this is indeed the case.

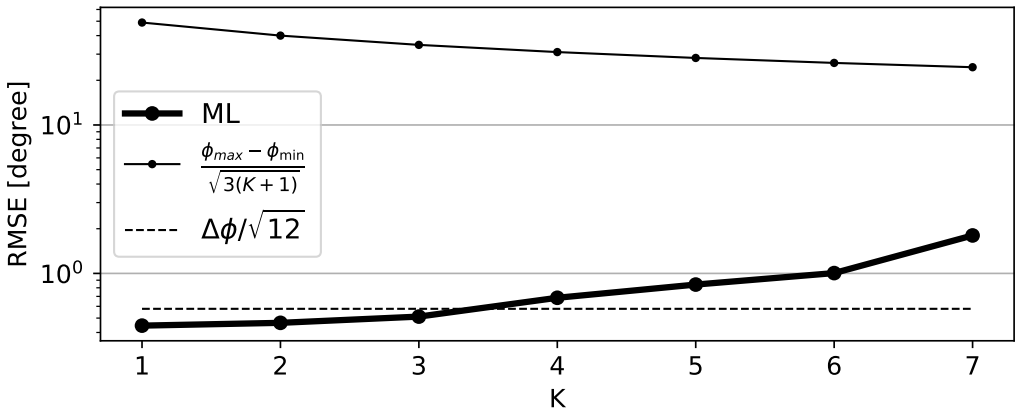


Figure 4.12: RMSE vs. K, SNR = 15 dB.

It is observed that the RMSE increases monotonically with the number of signals K, from 0.4° for $K = 1$ to 1.8° for $K = 7$. If $K \leq 3$, the RMSE is below $\Delta\phi/\sqrt{12} \approx 0.58°$, i.e. the lower bound for perfect estimators in a framework without border perturbations. The graph $(\phi_{max} - \phi_{min})/\sqrt{3(K+1)}$ indicates the RMSE that would be obtained when the K DOAs were estimated by K uniformly distributed random variables (appendix D.1), meaning that it can be interpreted as an upper bound. it can be seen that the RMSE of the DOA estimates performed by the ML framework is at least one order of magnitude smaller than this upper limit.

Combining all information from this section it can be concluded that the performance of the framework, in terms of all considered metrics, decreases with K for observations of a 15 dB SNR. For the metric $P(\hat{K} = K)$ this was expected because of the finite resolution that comes with the framework. However, it is shown that for those observations for which $\hat{K} = K$, the RMSE increases with K as well, whereas it would be constant over the various values of K if the finite resolution would be the only cause of a decreasing $P(\hat{K} = K)$. Nevertheless, the RMSE is below the minimum RMSE for perfect classifiers in a framework without border perturbations if $K \leq 3$. As $P(\hat{K} = K)$ equals 100% if $K = 1$, it is once more confirmed that the border perturbations benefit the estimates.

Despite the uniform distribution of K over all observations in the training set, the performance of the classifier ensemble in terms of both precision and recall decreases with K. A possible explanation for this phenomenon could be the following. The number of grid segments is defined as M and the number of signals impinging the array is called K. Assuming each of the K signals is associated with a different grid segment, the number of different combinations of K of those M segments is $\binom{M}{K}$. When $M = 60$ and $K = 2$, this yields 1770 different combinations. For a training set of 10^5 observations as was used in section 4.2.1, this implies that each

combination of segments appears, on average, 56.5 times. As a result, the NNs can learn features for each segment combination as a whole. When $K = 7$ though, 3.86×10^8 different segment combinations exist, such that learning the features for each combination becomes infeasible. Instead, the NNs must learn features for each individual grid segment and extract those features even if they are combined in a single observation. The theory above could be validated by, e.g., repeating the $K = 2$ scenario using a training set containing only a subset of the 1770 different combinations and a test set containing all combinations.

Chapter 5

Conclusions and recommendations

5.1 Conclusions

The main goal of this work was to devise a machine learning framework which is able to estimate the directions-of-arrival of an unknown number of signals. The framework proposed in Chapter 3 can be employed in combination with any learning algorithm capable of single-label multi-class classification. Furthermore, it does not require any knowledge about the number of signals impinging the antenna array prior to the estimation.

The framework was employed in combination with feedforward neural networks. It was shown that the source number is estimated correctly in 93% of the observations with SNRs of at least -5 dB, if 2 signals impinge the antenna array. In the remaining 7%, the DOAs were closely spaced compared to the the resolution of the framework, such that they could not be resolved. Furthermore, it was shown that the performance of the framework gradually decreases with an increasing source number: a 100% accuracy regarding the estimated number of sources, together with an 0.4° RMSE, is achieved if one signal impinges the array, whereas the accuracy decreases to 3.3% and the RMSE increases to 1.8° if 7 signals impinge the array. A significant part of this decrease in performance can be explained by the 2° resolution of the framework. However, it is shown that the predictions of the networks become less precise with an increasing source number as well. Further research is required to determine the cause of this phenomenon and to find out if it can be counteracted.

The underlying thought of the assignment was to investigate the advantages of approaching the well-known DOA estimation problem by means of machine-learning, compared to conventional approaches such as the MUSIC algorithm. For the learning strategy and the synthetic data considered here, it is observed that this depends on the resolution of the framework and the SNR of the observations. The ML framework with a 2° resolution outperforms MUSIC in terms of the probability of resolution of closely spaced sources at an SNR of -5 dB (100% resolution probability

at spacings of at least 7.4° vs. 11.6° for MUSIC), whereas the opposite applies for a 15 dB SNR (5.9° vs. 3.5°). After increasing the resolution of the framework to 0.8°, the MUSIC algorithm was outperformed at a 15 dB SNR as well (3.2°). However, the training set was increased by a factor 4 to achieve this. In other words, it seems that the amount of available data determines the maximum resolution that can be achieved for a certain learning strategy. Further research is required to establish the exact relation between them.

5.2 Recommendations

The peak detection algorithm which is applied to the spatial spectra could have been improved by taking the width of the peaks into account as well. In its current form, it cannot resolve DOAs of multiple signals when they are associated with neighbouring grid segments. Alternatively, the peak detector could be replaced by a machine learning technique such as a convolutional neural network. However, additional training data would be required to do this.

The simulations conducted in this work showed that class imbalance is inherent to the approach, and that this imbalance increases with the resolution of the framework. As a result, a larger training set is required such that the minority classes are represented by a sufficient amount of observations. It is recommended to investigate if the need for additional data can be diminished by using a different learning algorithm, or by applying advanced techniques for handling class imbalance such as oversampling or synthetic data generation [17].

The underlying thought of the assignment was to investigate if DOA estimation could benefit from utilizing machine learning, compared to conventional model-based methods. The data used to compare the ML framework and the MUSIC algorithm is created synthetically according to a well-known data model. However, this model is identical to the model on which the MUSIC algorithm is based. Comparing both methods using real-world data could give more insight in the advantages of a data-driven approach. Alternatively, effects such as array imperfections and mutual coupling could be included in the data-model.

Another future investigation could be to devise a DOA estimation evaluation metric which combines both the source number as well as the angular estimation errors. The probability of resolution is an example of such a metric. However it is only defined for two sources, and it equals 0 as soon as the estimated number of sources does not equal two. Furthermore, it relates the DOA estimates to the Rayleigh resolution limit, a limit which is overcome by both the ML framework as well as the MUSIC algorithm.

Bibliography

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

[2] R. Roy and T. Kailath, “Esprit-estimation of signal parameters via rotational invariance techniques,” *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.

[3] Z. Yang, J. Li, P. Stoica, and L. Xie, “Sparse methods for direction-of-arrival estimation,” in *Academic Press Library in Signal Processing, Volume 7*. Elsevier, 2018, pp. 509–581.

[4] M. Wax and T. Kailath, “Detection of signals by information theoretic criteria,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 387–392, 1985.

[5] A. Khan, S. Wang, and Z. Zhu, “Angle-of-arrival estimation using an adaptive machine learning framework,” *IEEE Communications Letters*, vol. 23, no. 2, pp. 294–297, 2018.

[6] Y. Kase, T. Nishimura, T. Ohgane, Y. Ogawa, D. Kitayama, and Y. Kishiyama, “Doa estimation of two targets with deep learning,” in *2018 15th Workshop on Positioning, Navigation and Communications (WPNC)*. IEEE, 2018, pp. 1–5.

[7] Z.-M. Liu, C. Zhang, and S. Y. Philip, “Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections,” *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315–7327, 2018.

[8] M. Pastorino and A. Randazzo, “A smart antenna system for direction of arrival estimation based on a support vector regression,” *IEEE transactions on antennas and propagation*, vol. 53, no. 7, pp. 2161–2168, 2005.

[9] O. Bialer, N. Garnett, and T. Tirer, “Performance advantages of deep neural networks for angle of arrival estimation,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3907–3911.

[10] M. Agatonovic, Z. Stankovic, I. Milovanovic, N. S. Doncov, L. Sit, T. Zwick, and B. Milovanovic, “Efficient neural network approach for 2d doa estimation based on antenna array measurements,” *Progress In Electromagnetics Research*, vol. 137, pp. 741–758, 2013.

[11] M. Li and Y. Lu, “Angle-of-arrival estimation for localization and communication in wireless networks,” in *2008 16th European Signal Processing Conference*. IEEE, 2008, pp. 1–5.

[12] F. Chollet, *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.

[13] M.-L. Zhang and Z.-H. Zhou, “A review on multi-label learning algorithms,” *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[14] G. Tsoumakas and I. Vlahavas, “Random k-labelsets: An ensemble method for multilabel classification,” in *European conference on machine learning*. Springer, 2007, pp. 406–417.

[15] G. Tsoumakas, I. Katakis, and I. Vlahavas, “Random k-labelsets for multilabel classification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2010.

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[17] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 4, pp. 463–484, 2011.

[18] H. L. Van Trees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

[19] K. I. Pedersen, P. E. Mogensen, and B. H. Fleury, “Power azimuth spectrum in outdoor environments,” *Electronics Letters*, vol. 33, no. 18, pp. 1583–1584, 1997.

[20] A. F. Molisch, *Wireless communications*. John Wiley & Sons, 2012, vol. 34.

Appendix A

Performance metrics

A.1 Classification performance

Evaluating the performance of a classifier can be done in many different ways. A well-known metric is the accuracy, which is defined as the number of correct predictions divided by the total number of predictions. It can therefore be computed for both binary as well as multi-class classification problems. For a binary problem, it can be expressed in terms of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn) as follows

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn}. \tag{A.1}$$

When dealing with a classification problem with a significant class imbalance, a high accuracy can be obtained by assigning the label of the majority to each new observation. However, such an estimator would have no statistical power. In such a case, more insight can be gained by evaluating the classifier for each class individually. This could be done by means of, e.g., the metrics precision, recall and F_β-score.

The precision is a measure for the classifiers exactness and tells how many of the positive estimates are actually correct. It is defined as

$$precision = \frac{tp}{tp + fp}. \tag{A.2}$$

The recall is a measure for the classifiers completeness, and describes how many of the observations that should have been predicted positive, are actually predicted as such. It is defined as

$$recall = \frac{tp}{tp + fn}. \tag{A.3}$$

In practice, improving one of the above is at the expense of the other. For example, a recall of 1 can easily be obtained by assigning the 'True'-label to each observation.

However, this will increase the number of false positives and therefore decrease the precision.

The F_β-score is the harmonic mean of precision and recall. The relative weighting of the two can be modified via the parameter β, where $\beta = 1$ implies that both are taken into account equally much. The F_β-score is defined as

$$\begin{aligned} F_\beta &= (1+\beta^2)\frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall} \\ &= \frac{(1+\beta^2)tp}{(1+\beta^2)tp + \beta^2 fn + fp}. \end{aligned} \tag{A.4}$$

To express the performance of a multi-class classifier in a single number, an average over the different classes is to be performed. Two well-known methods that can be used to compute this average are the micro and the macro average. For some binary performance evaluation function $B(tp, tn, fp, fn)$, e.g. precision, recall or F_β-score, the micro average of that function, B_{micro}, is computed as:

$$B_{micro} = B\left(\frac{1}{C}\sum_{i=1}^{C} tp_i, \frac{1}{C}\sum_{i=1}^{C} tn_i, \frac{1}{C}\sum_{i=1}^{C} fp_i, \frac{1}{C}\sum_{i=1}^{C} fn_i\right), \tag{A.5}$$

where C is the number of classes in the multi-class problem. The macro average is computed as

$$B_{macro} = \frac{1}{C}\sum_{i=1}^{C} B(tp_i, tn_i, fp_i, fn_i). \tag{A.6}$$

It can be seen that the support of the different classes is implicitly taken into account by the micro average, whereas all classes contribute equally much to the macro average, independent of their support.

A.2 RMSE

Consider Q observations for which a certain number of directions-of-arrival (DOAs) is to be estimated. This number, K_q with $q = 1, \ldots, Q$, could vary over the observations. The root-mean-square error (RMSE) is computed by taking the square root of the average of the mean-square error (MSE) of all separate observations:

$$\mathrm{RMSE} = \sqrt{\frac{1}{Q}\sum_{q=1}^{Q}\left[\frac{1}{K_q}\sum_{k=1}^{K_q}(\phi_{k,q} - \hat{\phi}_{k,q})^2\right]}. \tag{A.7}$$

Here, $\phi_{k,q}$ is the k^{th} DOA of test q and $\hat{\phi}_{k,q}$ is the k^{th} DOA estimate of test q, with $\phi_{1,q} < \cdots < \phi_{K_q,q}$ and $\hat{\phi}_{1,q} < \cdots < \hat{\phi}_{K_q,q}$. The RMSE can only be computed if $\hat{K}_q = K_q$, i.e. if the number of estimated DOAs equals the true number of DOAs.

A.3 Probability of resolution

The probability of resolution evaluates the quality of the direction-of-arrival (DOA) estimates by comparing them to both the true DOAs and the Rayleigh resolution limit. This limit is defined as $BW_{nn}/2$, with BW_{nn} being the null-to-null beamwidth of the array. For an N-element ULA of spacing d and wavelength λ, BW_{nn} is defined as [18]

$$BW_{nn} = 2\sin^{-1}\left(\frac{\lambda}{Nd}\right). \tag{A.8}$$

The Rayleigh resolution limit is a measure for the capability of an array to resolve two signals [18]. Conventional non-parametric methods such as the beamscan method can only resolve signals for which the DOA is spacing is larger than $BW_{nn}/2$.

The probability of resolution [18] is defined in u-space. A certain DOA ϕ_k and its estimate $\hat{\phi}_k$ can be expressed in u-space as

$$u_k = \cos\left(\phi_k\right) \tag{A.9a}$$

$$\hat{u}_k = \cos\left(\hat{\phi}_k\right). \tag{A.9b}$$

If two signals, i.e. $k \in \{1, 2\}$, impinge an ULA of N elements with element spacing d, they are considered to be resolved if the following conditions hold:

$$|\hat{u}_1 - u_1| \leq \min\left(\frac{u_2 - u_1}{2}, \frac{\lambda}{Nd}\right) \tag{A.10a}$$

$$|\hat{u}_2 - u_2| \leq \min\left(\frac{u_2 - u_1}{2}, \frac{\lambda}{Nd}\right). \tag{A.10b}$$

Appendix B

Benchmarks

B.1 MDL and AIC

The number of signals impinging a sensor array can be estimated using, e.g., a subspace order estimator. Two examples are the minimum description length (MDL) and Akaike's information criterion (AIC). [4]. Both are based on the eigenvalues of the estimated sensor covariance matrix, $\lambda_1, \ldots, \lambda_N$, with $\lambda_1 > \cdots > \lambda_N$. By definition, both methods are limited to estimating at most $N-1$ signals, with N being the number of elements in the array. The estimated number of signals $\hat{K}$ is computed as

$$\hat{K}_{\mathrm{MDL}} = \arg\min_k \mathrm{MDL}(k) \tag{B.1a}$$

$$\hat{K}_{\mathrm{AIC}} = \arg\min_k \mathrm{AIC}(k) \tag{B.1b}$$

where the subscribt of $\hat{K}$ indicates the method used for the estimation and $k = 0, 1, \ldots, N-1$. The functions $\mathrm{MDL}(k)$ and $\mathrm{AIC}(k)$ are given by

$$\mathrm{MDL}(k) = f(k) + k(2N-k)\frac{1}{2}\ln(T) \tag{B.2a}$$

$$\mathrm{AIC}(k) = f(k) + k(2N-k), \tag{B.2b}$$

respectively, where N denotes the number of antenna elements, T is the number of snapshots and

$$f(k) = -T(N-k)\ln\left(\frac{\left[\prod_{n=k+1}^{N}\lambda_n\right]^{\frac{1}{N-k}}}{\frac{1}{N-k}\sum_{n=k+1}^{N}\lambda_n}\right). \tag{B.3}$$

B.2 Cramér-Rao lower bound

The Cramér-Rao lower bound (CRLB) is the minimum variance of an unbiased estimator. When applied to the 1D DOA estimation of K signals, it is a square matrix $\mathbf{CRLB} \in \mathbb{R}^{K\times K}$, of which the k^{th} diagonal element represents the minimum variance of the estimate of the k^{th} signal impinging the array. In this work the stochastic CRLB is considered, as it is assumed that both signals and noise are random Gaussian processes. The CRLB is computed as [18]

$$\mathbf{CRLB} = \frac{\nu^2}{2T}\Big[\Re\big\{(\mathbf{D}^H\mathbf{\Pi}_{\mathbf{A}}^{\perp}\mathbf{D}) \odot (\mathbf{P}\mathbf{A}^H\mathbf{R}^{-1}\mathbf{A}\mathbf{P})^T\big\}\Big]^{-1}, \tag{B.4}$$

where $\Re\{\cdot\}$ denotes the real part(s) of a complex-valued entry and $\odot$ is the Hadamard product. Furthermore, ν^2 is the noise variance, T is the number of available snapshots and $\mathbf{A} \in \mathbb{C}^{N\times K}$ is the array manifold matrix consisting of steering vectors $\mathbf{a}_1, \ldots, \mathbf{a}_K$, with N being the number of antenna elements in the array. $\mathbf{D} \in \mathbb{C}^{N\times K}$ is the derivative of the array manifold matrix with respect to the K DOAs:

$$\mathbf{D} = \begin{bmatrix} \dfrac{d\mathbf{a}_1}{d\phi_1} & \dfrac{d\mathbf{a}_2}{d\phi_2} & \cdots & \dfrac{d\mathbf{a}_K}{d\phi_K} \end{bmatrix}. \tag{B.5}$$

Furthermore, $\mathbf{P} \in \mathbb{C}^{K\times K}$ is the signal covariance matrix and $\mathbf{R} \in \mathbb{C}^{N\times N}$ is the sensor covariance matrix which is computed as

$$\mathbf{R} = \mathbf{A}\mathbf{P}\mathbf{A}^H + \nu^2\mathbf{I}_N \tag{B.6}$$

with $\mathbf{I}_N$ being the $N \times N$ identity matrix. The projection matrix onto the noise subspace $\mathbf{\Pi}_{\mathbf{A}}^{\perp}$ is given by

$$\mathbf{\Pi}_{\mathbf{A}}^{\perp} = \mathbf{I}_N - \mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H. \tag{B.7}$$

The square root of the CRLB can be compared to the RMSE (appendix A.2) of the DOA estimates.

B.3 MUSIC

The multiple signal classification (MUSIC) algorithm [1] is a subspace-based super-resolution method which is based on the eigendecomposition of the estimated sensor covariance matrix $\hat{\mathbf{R}}$. Before being able to compute the MUSIC spectrum, the number of signals K impinging the N-element array should be known. The eigenvectors associated with the $N-K$ smallest eigenvalues together form the estimated noise subspace $\mathbf{E}_n \in \mathbb{C}^{N\times(N-K)}$. The MUSIC spectrum P_{MUSIC} is evaluated at scanangle ϕ according to

$$P_{\text{MUSIC}}(\phi) = \frac{1}{\mathbf{a}^H(\phi)\mathbf{E}_n\mathbf{E}_n^H\mathbf{a}(\phi)}, \tag{B.8}$$

where $\mathbf{a}(\phi)$ is the steering vector for direction ϕ.

Due to orthogonality of the noise and signal subspace, the denominator in Eq. (B.8) theoretically equals 0 when $\phi = \phi_k$, with ϕ_k being the direction-of-arrival (DOA) of the k^{th} signal for $k = 1, \ldots, K$. This results in peaks in the spectrum at those specific angles. In practice, the sensor covariance matrix $\mathbf{R}$ and the number of incident signals K are estimated from noisy measurement data, meaning that the $\mathbf{E}_n$ will be less accurate for lower signal-to-noise ratios.

Note that the estimated DOAs are part of a discrete set of scan-angles at which the MUSIC spectrum is evaluated, meaning that the estimator has a finite resolution.

Appendix C

Additional results

C.1 Results for two sources

The results in this section were obtained using a resolution of $\Delta\phi = 2^\circ$, with a border perturbation of at most 0.5°. Two signals of which the directions-of-arrival (DOAs) are drawn from $U(30^\circ, 150^\circ)$ impinge the array, which is uniform linear array of 8 elements with half wavelength spacing.

C.1.1 Size of the training set

In this appendix, the impact of the amount of training observations, from now on referred to as Q_{train}, on the performance of the ML framework is investigated. The performance of the framework will be both expressed in terms of classification evaluation metrics as well as in relevant metrics in the field of DOA estimation. Four simulations were conducted in which Q_{train} was varied between 2.5×10^4 and 2×10^5 observations. The test set consists of $Q_{test} = 2.2 \times 10^4$ test experiments in total: 2×10^3 observations per SNR, with the SNRs being $-20, -15, \ldots, 30$ dB.

A significant class imbalance is inherent to the framework: given the settings of the considered scenario, i.e. $M = 120^\circ/2^\circ = 60$, $k = 3$ and $K = 2$, it is expected that $(57/60)^2 \times 100\% = 90.25\%$ of the observations in both train and test set is associated with $K_{NN} = 0$ (appendix D.4), with K_{NN} being the number of positive labels associated with a certain class. Because of the class imbalance, evaluating the performance of the classifier ensemble is preferably done for each class individually. As the well-known accuracy is ill-defined for a single class of a multi-class problem, it was decided to evaluate the classification performance using the metrics precision, recall and F_1-score. In order to express the classifier performance in a single number, some way to average over the different classes is required. The micro and macro average are two different approaches to do this. The definitions of these metrics and averages can be found in appendix A.1.

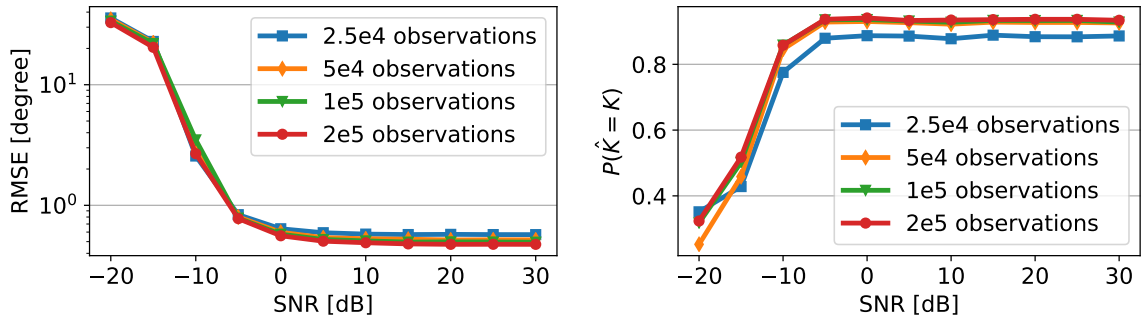
Table C.1 shows the micro and macro average of all metrics mentioned above for the 4 different values of Q_{train}. Note that the numbers shown in the table are averaged over all NNs in the classifier ensemble.

Table C.1: Classification metrics vs. training set size, averaged over all NNs

Q_{train}	precision (%)		recall (%)		F_1-score (%)	
$(\times 10^5)$	micro	macro	micro	macro	micro	macro
0.25	93.1 ± 0.6	53.3 ± 11.2	93.1 ± 0.6	42.9 ± 9.4	93.1 ± 0.6	45.5 ± 9.0
0.5	94.0 ± 0.6	67.3 ± 10.3	94.0 ± 0.6	49.9 ± 9.0	94.0 ± 0.6	57.0 ± 8.6
1	94.8 ± 0.4	73.6 ± 7.9	94.8 ± 0.4	60.5 ± 7.6	94.8 ± 0.4	64.6 ± 6.7
2	95.2 ± 0.4	76.7 ± 6.4	95.2 ± 0.4	64.0 ± 6.2	95.2 ± 0.4	68.3 ± 5.5

From the data presented in Table C.1, it can be concluded that the performance of the classifier ensemble increases with the amount of observations in the training set. For example, the macro F_1 increases by about 23% when using the largest training set instead of the smallest one. The micro F_1 is 93% for the smallest data set and increases up to 95% for $Q_{train} = 2 \times 10^5$. The difference of at least 27% between micro and macro F_1 is a result of the class imbalance, as the support of the different classes is taken into account only by the micro average.

Due to the adaptive threshold in the framework in combination with the fact that each segment of the grid is covered by multiple classifiers, a higher classification performance does not necessarily imply an improved RMSE and/or $P(\hat{K} = K)$. Both metrics are shown in Fig. C.1.



(a) RMSE vs. SNR *(b) $P(\hat{K} = k)$ vs. SNR*

Figure C.1: ML framework DOA estimation performance for various Q_{train}, 2×10^3 test observations per SNR.

In Fig. C.1a, it can be observed that the RMSE decreases with an increasing SNR for all values of Q_{train}. For SNRs larger than 5 dB, the RMSE stabilizes. Increasing Q_{train} from 2.5×10^4 to 2×10^5 decreases the RMSE by at most 0.3° for SNRs larger than -5 dB. For the same SNRs and the same increase in Q_{train},

$P(\hat{K} = K)$ increases from 88% to 93% (Fig. C.1b). The difference in $P(\hat{K} = K)$ for $Q_{train} = 1 \times 10^5$ and $Q_{train} = 2 \times 10^5$ is less than 1% for all SNRs except -15 dB (1.6%). For this specific scenario, training on 10^5 observations is therefore considered sufficient. Note that this might change as soon as parameters such as the resolution $\Delta\phi$, the layout of the NNs, the number of signals K etc. change.

C.1.2 Classifier performance

The results presented in this paragraph were obtained using a NNs ensemble trained on $Q_{train} = 10^5$ observations. The test set consists of 1.5×10^4 observations for each SNR $\in \{-20, -15, \ldots, 30\}$ dB, resulting in $Q_{test} = 1.65 \times 10^5$ test experiments in total.

The DOA estimation performance of the ML framework depends on the classification performance of the NNs in the ensemble, i.e. their ability to correctly choose 1 out of $2^k = 2^3 = 8$ classes for each observation. The latter can be evaluated by means of, e.g. a confusion matrix, which is shown in Fig. C.2. The numbers in the matrix represent the amount of observations corresponding to each target-estimate pair. Note that numbers between 0 and 1 are possible, as the matrix is averaged over all NNs in the ensemble. The colors represent the same information. The grid is added such that clusters of K_{NN}, i.e. the amount of positive labels, are clarified. As the number of sources $K = 2$, the class (111) never occurs.

Two conclusions are drawn from this confusion matrix. One is about the usefulness of the adaptive threshold in the peak detection algorithm, and the other is about the metrics which can be used to assess the performance of the classifiers in a more compact way.

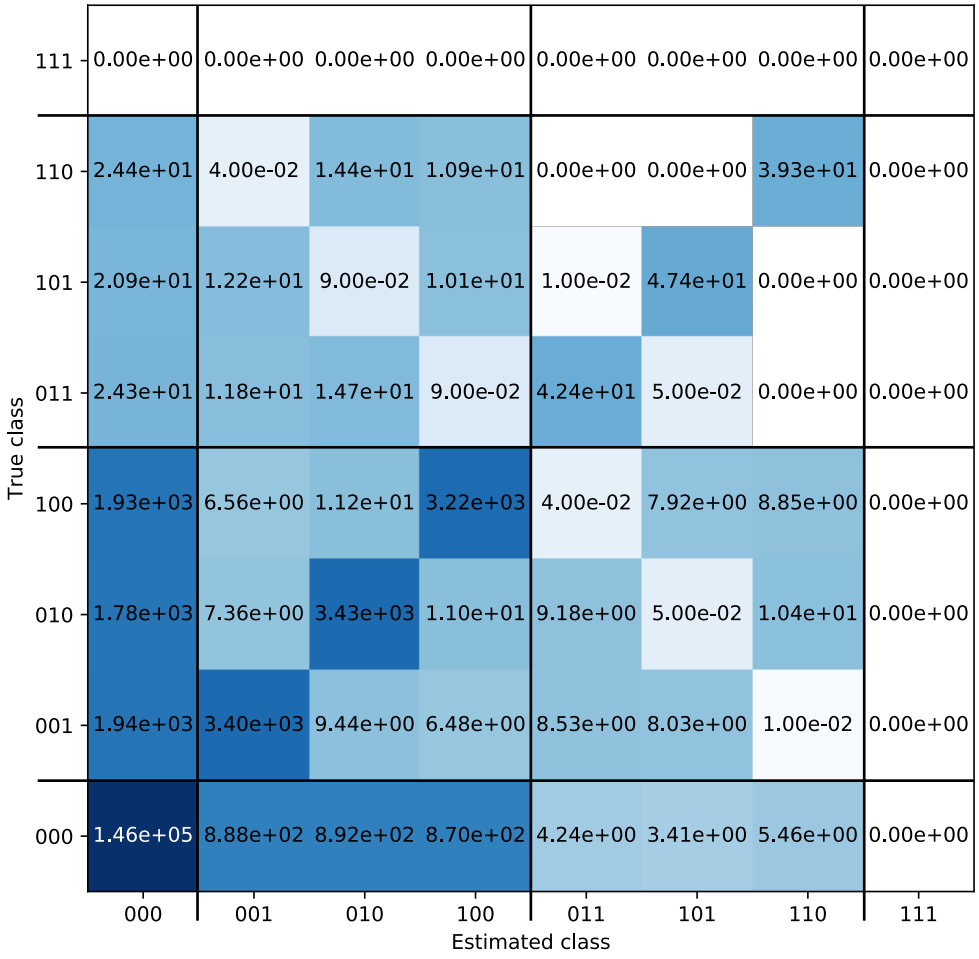


Figure C.2: Confusion matrix of true and estimated classes

Reduced height of spectrum peaks

Looking at the confusion matrix in Fig. C.2, it can be seen that for true classes corresponding to $K_{NN} = 1$, i.e. (001), (010) and (100), the ML framework either predicts the correct class or it predicts the class (000). Other classes are estimated at least 2 orders of magnitude less frequent.

For true classes associated with $K_{NN} = 2$, i.e. (011), (101) and (110), it predicts the correct class for roughly half of the observations. Class (011) is evaluated as an example:

$$\frac{4.24}{4.24 + 2.43 + 1.18 + 1.47} \approx 0.45, \tag{C.1}$$

i.e. of all observations associated with true class (011), 45% is estimated as such. In other words, the classifier ensemble is 45% complete for class (011). The remaining 55 % is either predicted as class (000), or as a class associated with $K_{NN} = 1$. The latter only applies to those classes for which the label that is predicted as being true, is one of the two labels which are actually true. For example, when rounding $9.00e^{-2}$ to an integer amount of observations, i.e. 0, it can be seen that observations of class (011) are never predicted as (100).

From the numbers in the confusion matrix, it can be concluded that for the given scenario of 2 impinging signals, the DOA estimation relies heavily on classes associated with $K_{NN} = 1$. Most incorrect predictions for observations from these classes imply that a label is not assigned whereas it should have been, rather than the opposite. Whereas the former only results in spectrum peaks which are lower than they would have been with perfect classifiers, the latter would result in peaks at unwanted angles. Due to the adaptive threshold in the peak detection algorithm, the reduced peak height is taken into account.

Performance evaluation metrics

The confusion matrix presented in Fig. C.2 shows a significant class imbalance. When adding the numbers in the matrix row-wise, it can be seen that roughly 90% (about 1.49×10^5 observations) of all 1.65×10^5 observations correspond to class (000). As 1.46×10^5 of the 1.49×10^5 observations of class (000) are estimated as such, the accuracy of the classifier ensemble will be at least $1.46/1.65 \times 100\% = 88.5\%$. Only considering the accuracy of the classifier ensemble is misleading, as it will be dominated by class (000). The performance of the classier network should therefore be evaluated for each class individually. The accuracy for a single class of a multi-class problem is ill-defined (should you evaluate the matrix row-wise or column-wise?), meaning that different metrics are to be used.

Examples of other metrics which give a more compact representation of the clas-

sifier performance are precision, recall and F_1-score (appendix A.1). These metrics, averaged over all NNs in the ensemble, are presented Table C.2. The relative support, defined as the absolute support divided by Q_{test}, is given as well. The classes are clustered based on K_{NN}, i.e. the number of true labels (note that each class represents $k = 3$ labels, i.e. grid segments). If multiple classes are associated with a certain K_{NN}, a macro average is computed for those classes for each NN individually. Afterwards, a single mean and standard deviation is computed over all macro averages. Note that, as the number of signals impinging the array $K = 2$, the support of $K_{NN} = 3$ equals 0, such that no metrics can be computed.

Table C.2: Classification metrics per class, averaged over all NNs, $Q_{train} = 10^5$, $Q_{test} = 1.65 \times 10^5$, $\Delta\phi = 2°$

K_{NN}	class	precision (%)	recall (%)	F_1-score (%)	support (%)
0	(000)	96.2 ± 0.4	98.2 ± 0.4	97.2 ± 0.2	90.3 ± 1.2
1	(001) (010) (100)	78.3 ± 5.9	62.5 ± 8.5	69.1 ± 6.2	3.2 ± 0.7
2	(011) (101) (110)	65.2 ± 19.7	45.5 ± 18.8	51.3 ± 17.4	$(6.0 \pm 1.8)e^{-2}$
3	(111)	-	-	-	0

The relative support shown in Table C.2 indicates a significant class imbalance, as 90.3% of the observations is associated to class (000). It can be seen that the support decreases with increasing K_{NN}. Note that the relative support is the same for the train and the test set for the simulation scenario considered here. In other words, the bigger K_{NN}, the fewer examples present in the training set from which the NNs could learn. This explains why the precision, recall and F_1-score decrease with increasing K_{NN}.

C.1.3 Average DOA spacing

In Fig.C.3, the average direction-of-arrival (DOA) spacing of all observations for which the estimated number of sources is incorrect is plotted against the signal-to-noise ratio (SNR). For SNRs of -20 dB and -15 dB, the average spacing equals 39.8° and 38.8° respectively, which is close to the expected average spacing between two random, uniformly distributed, DOAs in the domain of $[30°, 150°]$, i.e. 40° (appendix D.2). However, for SNRs of -5 dB and higher, the average spacing is at most 2.8°.

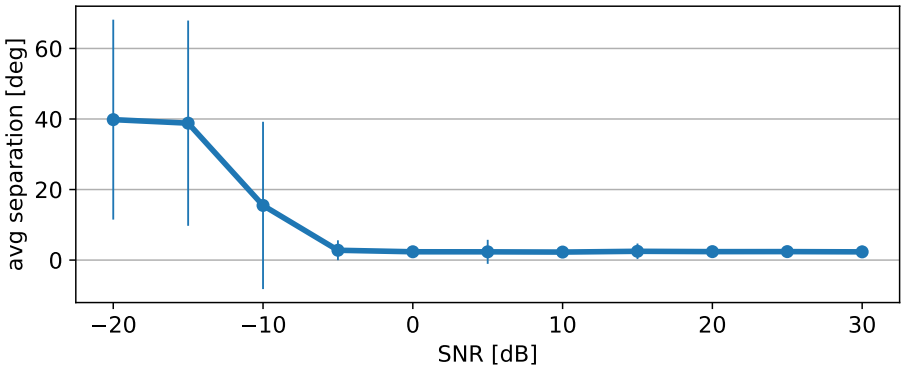


Figure C.3: Average DOA spacing of observations for which $\hat{K}_{ML} \neq 2$

C.1.4 Border perturbations

Perturbing the borders of the grid by at most 0.5° benefits the root-mean-square error (RMSE) of the direction-of-arrival estimates performed by the machine-learning framework. The value $\Delta\phi/\sqrt{(12)}$ is the theoretical lower bound for perfect classifiers in a framework without perturbations (appendix D.3).

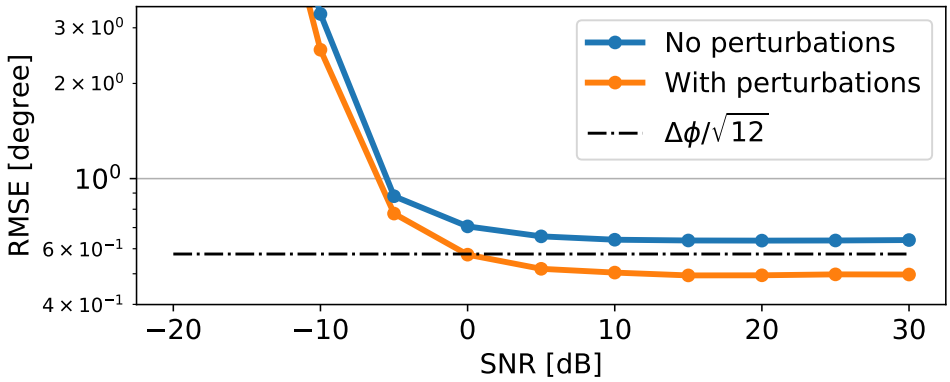


Figure C.4: RMSE vs. SNR, with and without border perturbations.

C.2 Results for varying number of sources

The results in this section were obtained using a resolution of $\Delta\phi = 2°$, with a border perturbation of at most 0.5°. $K = 1, \ldots, 7$ signals of which the directions-of-arrival (DOAs) are drawn from $U(30°, 150°)$ impinge the array, which is uniform linear array of 8 elements with half wavelength spacing.

C.2.1 Confusion matrix

The confusion matrix presented in Fig. C.5 corresponds to a signal-to-noise ratio (SNR) of 15 dB.

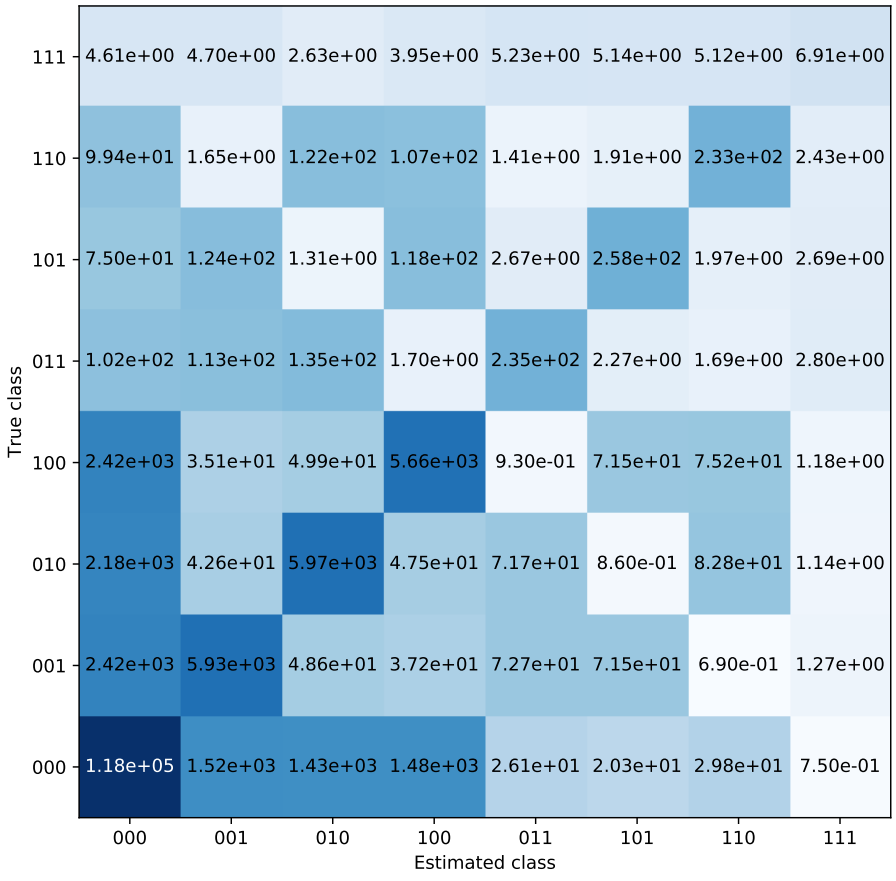


Figure C.5: Confusion matrix of true and estimated classes

Appendix D

Mathematical derivations

D.1 RMSE for random DOA estimates

Consider Q observations in which K signals impinge an antenna array. The K directions-of-arrival (DOAs) are uniformly distributed random variables in the domain $[\phi_{min}, \phi_{max}]$. Assume the DOAs are estimated by drawing K times (for each observation) from that same uniform distribution. The RMSE of these estimates, computed as in appendix A.2, will converge to a constant when Q approaches infinity. This constant can be computed using order statistics.

The sorted DOAs, i.e. $\phi_1 < \cdots < \phi_K$, and their estimates, $\hat{\phi}_1 < \cdots < \hat{\phi}_K$, are Beta-distributed random variables. The pdf of the Beta distribution f_{Beta} is given by

$$f_{\mathrm{Beta}}(x; \alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{\mathrm{B}(\alpha, \beta)}. \tag{D.1}$$

Here, $\mathrm{B}(\alpha, \beta)$ given by

$$\mathrm{B}(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \tag{D.2}$$

and $\Gamma(\cdot)$ is the Gamma function.

The squared error between a DOA ϕ and its estimate $\hat{\phi}$ is written as a function $g(\phi, \hat{\phi})$, which is defined as

$$g(\phi, \hat{\phi}) = (\phi - \hat{\phi})^2. \tag{D.3}$$

Furhtermore, the width $\phi_{max} - \phi_{min}$ of the considered domain is called Φ. The MSE of the estimate of the k^{th} signal, MSE_k that would be obtained when averaging over infinite observations is computed as

$$\mathrm{MSE}_k = \frac{1}{\Phi^2} \int_0^{\Phi} \int_0^{\Phi} g(\phi_k, \hat{\phi}_k) f_{\mathrm{Beta}}\Big(\frac{\phi_k}{\Phi}; \alpha_k, \beta_k\Big) f_{\mathrm{Beta}}\Big(\frac{\hat{\phi}_k}{\Phi}; \alpha_k, \beta_k\Big) d\phi_k d\hat{\phi}_k \tag{D.4}$$

with $\alpha_k = k$ and $\beta_k = K + 1 - k$. Finally, the RMSE is computed as

$$\mathrm{RMSE} = \sqrt{\frac{1}{K} \sum_{k=1}^{K} \mathrm{MSE}_k} \tag{D.5}$$

Using the symbolic toolbox of MATLAB, it was found that this equals

$$\mathrm{RMSE} = \frac{\Phi}{\sqrt{3(K+1)}} \tag{D.6}$$

D.2 Average spacing between neighbouring random DOAs

Consider Q observations in which K signals impinge an antenna array. The directions-of-arrival (DOAs) are uniformly distributed random variables in the domain $[\phi_{min}, \phi_{max}]$. The average spacing d_{avg} between two neighbouring DOAs will converge to a constant when Q approaches infinity. This constant can be computed in a way similar to the derivation presented in appendix D.1. By defining the function $g(\cdot)$ as

$$g(\phi_1, \phi_2) = \phi_2 - \phi_1 \tag{D.7}$$

with $\phi_2 > \phi_1$ and $\Phi = \phi_{max} - \phi_{min}$, it follows that

$$d_{avg} = \frac{1}{\Phi^2} \int_0^{\Phi} \int_0^{\Phi} g(\phi_k, \phi_{k+1}) f_{\mathrm{Beta}}\Big(\frac{\phi_k}{\Phi}; \alpha_k, \beta_k\Big) f_{\mathrm{Beta}}\Big(\frac{\phi_{k+1}}{\Phi}; \alpha_{k+1}, \beta_{k+1}\Big) d\phi_k d\phi_{k+1}. \tag{D.8}$$

Here, $k = 1, \ldots, K-1$, $\alpha_i = i$, $\beta_i = K+1-i$ and f_{Beta} is as defined in appendix D.1. Using the symbolic toolbox of MATLAB, it was found that

$$d_{avg} = \frac{\Phi}{K+1}. \tag{D.9}$$

Note that d_{avg} is independent of k.

D.3 RMSE for ideal classifiers without border perturbations

Assume a signal impinges a sensor array from direction-of-arrival (DOA) ϕ, which is a random variable of the uniform distribution. A perfect classifier with a resolution of $\Delta\phi$, applied to the observation described above, will result in an estimation error

$$\tilde{\phi} = |\phi - \hat{\phi}| \sim U(0, \Delta\phi/2) \tag{D.10}$$

with $\hat{\phi}$ being the DOA estimate.

The root-mean-square error (RMSE), defined as in appendix A.2, for an infinite amount of observations, converges to a constant. This constant is computed via the

mean-square-error according to

$$\mathrm{MSE} = \int_0^{\Delta\phi/2} \tilde{\phi}^2 \frac{1}{\Delta\phi/2} d\tilde{\phi} \tag{D.11a}$$

$$= \frac{1}{3\Delta\phi/2} \tilde{\phi}^3 \Big|_0^{\Delta\phi/2} \tag{D.11b}$$

$$= \frac{\Delta\phi^2}{12} \tag{D.11c}$$

The RMSE thus equals

$$\mathrm{RMSE} = \sqrt{MSE} \tag{D.12a}$$

$$= \frac{\Delta\phi}{\sqrt{12}} \tag{D.12b}$$

This applies to any number of signals K, as long as their DOAs are independent and identically distributed random varibles of the uniform distribution.

D.4 Expected relative support

Assume a framework consisting of M grid segments of equal width. Each classifier in the framework covers $k = 3$ of those segments. Furthermore, assume $K = 2$ signals impinge the antenna array, with random, uniformly distributed, directions-of-arrival (DOAs).

The $2^k = 8$ classes of the label powerset can be clustered in $k + 1 = 4$ classes, according to the total number of segments in which at least one signal impinges the array $k_{NN} = 0, \ldots, k$. The probability that an observation corresponds to a certain K_{NN} is computed as

$$P(K_{NN} = 0) = \left(\frac{M-3}{M}\right)^2 \tag{D.13a}$$

$$P(K_{NN} = 1) = 2\left(\frac{3}{M}\right)\left(\frac{M-3}{M}\right) + \frac{1}{M^2} \tag{D.13b}$$

$$P(K_{NN} = 2) = \left(\frac{3}{M}\right)\left(\frac{2}{M}\right) \tag{D.13c}$$

$$P(K_{NN} = 3) = 0. \tag{D.13d}$$