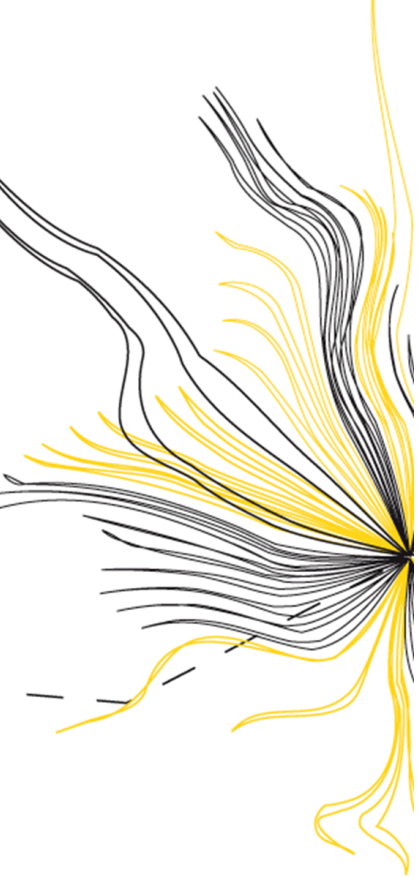


UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

THALES

An adaptive Bayesian approach towards a robust classifier for UAVs and birds

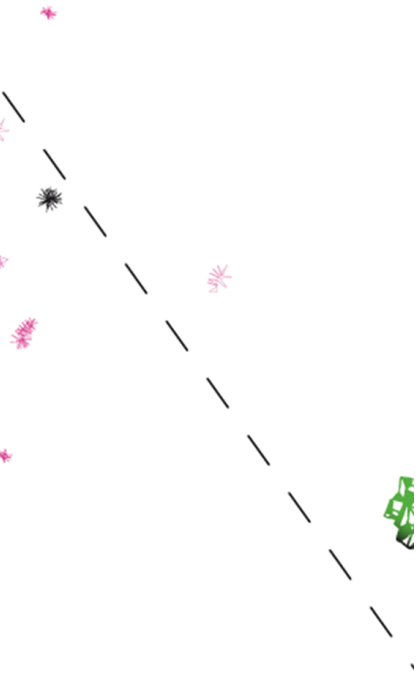
Ruben Heersink

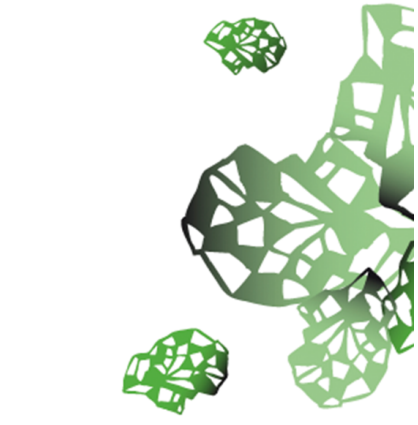
M.Sc. Thesis
September 2016

Assessment committee:
prof. dr. A.A. Stoorvogel
dr. P.K. Mandal
dr. M.A.E. Bocquel
dr. C. Brune

Supervisors:
dr. P.K. Mandal
dr. M.A.E. Bocquel

Chair: Hybrid Systems
Departement of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands





Abstract

In this thesis the problem of classification of birds and Unmanned Aerial Vehicles (UAVs) using a model-based Bayesian approach is considered. The conventional way of discriminating between birds and UAVs is based on the micro-Doppler signature which is induced by the micro motions of the target, such as the motion of wings and rotor blades for birds and UAVs respectively.

The model-based Bayesian approach is able to automatically classify targets and learn from experience. Hidden Markov models are developed based on the radar return model for the target and the associated class likelihood functions are derived. Maximum likelihood estimation is performed to estimate unknown parameters, which are subsequently used for classification. Unsupervised data are used to learn class dependent parameters by applying the learning technique called Maximum Likelihood Adaptive Neural System (MLANS). This approach does not require any preprocessing of the radar return signals and can simultaneously learn and classify. Moreover, the approach is robust with respect to uncertainties on parameter values, such as the initial position of the blades. The classification algorithm is tested on synthetic data and is shown to be capable to classify birds and UAVs with a 95% probability.

Keywords: UAV/Bird classification, Bayesian classification theory, Hidden Markov modelling, Learning, Maximum Likelihood Adaptive Neural Systems.

Acknowledgements

All good times come to an end. This Master's thesis means that my life as a student comes to an end and a new adventure is on its way.

This thesis is written during my final project in the Master programme Applied Mathematics at the University of Twente and I couldn't have done it by myself.

My first acknowledgement goes to my UT supervisor dr. Pranab Mandal for time he took for my questions, the detailed reviews of my thesis and the guidance throughout this final project. Secondly I want to thank UT/Thales supervisor dr. Melanie Bocquel for the guidance, detailed reviews of my work and encouraging talks.

I want to thank the members of my assessment committee prof. dr. Anton Stoorvogel and dr. Christoph Brune for the assessment of my work.

Further I would like to thank Yoeri Boink for the coffee breaks we had with encouraging discussions about our final projects. Last but not at least, I want to thank my family and Nina for the support during the final project.

Contents

Chapter 1

Introduction

1.1 Motivation

Over the last years, mini Unmanned Aerial Vehicles (UAVs) also known as drones have become more and more popular. Nowadays the prices are affordable, therefore accessability of the small flying helicopters has increased [36]. The UAVs are used for different non-commercial or commercial purposes. One might think of package delivery as a commercial purpose, but most UAVs that occupy the airspace are used for non-commercial purposes. A few examples of non-commercial usages of UAVs are UAVs equipped with cameras to get scenery videos of landmarks, UAVs used for racing or just as a toy to play around with.

One concern of this increased popularity is violation of privacy when people remotely control mini UAVs equipped with cameras. Another important concern is the abuse of UAVs for protests or criminal acts. UAVs can cause serious harm when equipped with an explosive device or used directly as an arm. Another concern is the usage of UAVs around airports. Lately there have been a lot of near-miss incidents with UAVs around airports, where airplanes (almost) collided with UAVs [13]. When considering UAVs as a potential threat for our security, suitable counteractions need to be developed. Therefore detection and identification of UAVs in the air space have become evermore important now threats are more serious due to the increase of accessibility to mini-UAVs.

Identification and classification of aerial vehicles is done by radar. Targets like UAVs are illuminated by radio waves and information about the target is extracted from the reflected signals. Even though the number of UAVs has increased over the past years, the most targets that are detected by the radar remain flying birds. Flying birds are not of interest for the radar operator, hence we want to filter out these detections and this is where automatic classification comes in. Automatic classification is a system that classifies the target and makes a decision whether the radar operator should be informed about this target or not. The classification between UAVs and birds is essential for the security issues, but besides the ability to distinguish between UAVs and birds it is also of interest to be able to classify the UAV in a specific subclass of UAVs. This last classification ability is interesting for military purposes.

This thesis deals with automatic classification of UAVs and birds for which an approach is presented.

1.2 Literature and contributions

Although the classification of mini-UAVs is quite a novel topic, the classification of other aerial vehicles using radar is not new. The conventional approach to extract information about a target is done by frequency analysis of the radar return signal.

A moving target causes a frequency shift in the carrier frequency of the transmitted signal. This is known as the Doppler effect [17]. If the target has any vibrating or rotating components, like a rotating propeller, a rotor of a helicopter or flapping wings of the birds, this will also induce a frequency modulation on the returned signal. These micro-motions cause micro-Doppler shifts. Frequency analysis of the radar return signal can give information about these components on a target which cause these micro-Doppler shifts. The micro-Doppler effect was originally introduced in the field of laser technology [38]. Later it was also used for radar application in [19], where the micro-motion of vibration was studied.

A more complete study of micro-Doppler effect in radar was done in [6]. Models for micro-Doppler frequency shifts were derived for vibration, rotation, tumbling and coning motions and were verified by simulation studies. The classification of helicopter by its micro-Doppler features was first investigate in [24]. In all this research, micro-Doppler features were extracted by time-frequency analysis tools, for instance the Short Time Fourier Transform (STFT). In the novel topic of classification of mini-UAVs these techniques were also used [25].

These frequency analysis tools induce a relative high computational cost due to the time-frequency transform and depend on the choice of parameters of this transform itself, for example the window length. These parameters depend on the dynamics of the target. A disadvantage of the time-frequency transform is loss of insight about how the parameters influence the classification and how the unknown parameters can be estimated.

A more direct approach was demonstrated in [22], where the theoretical radar return time signal was shown to depend on the number of blades, rotation speed and the length of the blades. The classification of helicopters based on this theoretical return was recently done in [15].

In this thesis classification of (mini-)UAVs and birds is also done based on the theoretical returns. To the best of our knowledge this is the first time that both (mini-)helicopters/UAVs and birds are classified based on the theoretical return. An advantage of using the theoretical return is that the model applies for all sensor settings/parameters, whereas the time-frequency based classifiers are built for one specific sensor setting. So this approach is robust to different sensor settings. The theoretical return classifier in [15] is built under the assumption that only measurement noise is present and is not robust to any unmodelled radar returns. The approach presented in this thesis is robust to possible unmodelled radar returns using a stochastic dynamical model. A stochastic model for the theoretical radar return is developed and used for classification. To our knowledge it is the first of its kind for this application. Another concept with this classification approach is that parameters in the underlying models are learned simultaneously with classification, this concept isn't used before in the field of UAV/bird classification.

1.3 Outline thesis

The remainder of the thesis is organised as follows.

In Chapter 2 background knowledge relevant this research work is presented. Firstly the general problem of classification is introduced. Several approaches solving the classification problem are then presented. One of these approaches is based on the statistical classification theory. The statistical classification theory is explained in more detail, in particular the Bayesian classification theory. Next hidden Markov modelling is discussed to deal with sequential data. Finally parameter learning and estimation techniques are investigated.

In Chapter 3 the main contribution of this thesis is presented. The received signal models for a single point scatterer on a UAV and a single point scatterer on a bird are derived. Next the associating stochastic dynamic class models and stochastic observation models are developed. Subsequently the hidden Markov models that arise from these dynamical and observations models are used to build the classifiers. Next the UAV and bird are modelled using multiple scatterers models, from which different hidden Markov models are derived and the corresponding classifier is presented.

In Chapter 4 the numerical results are presented. The performance of the classifier based on both single and multiple scatterer models is tested. Relaxations of assumptions are done to investigate the impact on the performance and the performance of parameter learning and estimations are discussed.

Finally, Chapter 5 concludes this thesis with the main conclusions from the analysis on the results in Chapter 4 and discusses the limitations of this study and gives directions for future research.

Chapter 2

Background knowledge

In this section background knowledge about classification theory and estimation theory is discussed. The general idea behind classification is presented and techniques solving the classification problem are investigated. The techniques that have more potential of solving the UAV/bird classification problem are discussed in more detail. First we explain the general task in classification.

The general task in classification is assigning a label/class/category to an object using the available information about its properties. It is a general problem in a wide range of fields [34]. Several general problems which are classification problems are recognition and detection.

In the recognition problem one can think of recognising a license plate on a photo, i.e. recognising the characters on the license plate. Ideally we want to assign each character on the photo with one of multiple labels: 'A', 'B',..., '0', '1'..'9', such that a license plate can be reconstructed.

The problem of detection is a binary classification problem with only two classes: 'Yes' and 'No', which is answering the question whether or not we are observing a certain phenomenon or object. For instance, we can detect targets in the air using a radar, hence the question is whether or not we are observing a target in the radar.

Classification is often learned from experience [3]. Given the experience, a set of samples for which the class is known, called labelled samples, the task is to learn how to classify new samples. The sample $y \in \mathcal{Y}$ is a set of attributes or features of an object and lies in the sample space $\mathcal{Y}$. The set of observations we use to learn experience from we call the training set. In a lot of classification problems the training samples/training set are a set of labelled observations so the class $c \in \mathcal{C}$ from which the sample y is originating is known, hence the classification system can be learned. Learning from such a training set can be done using different approaches or techniques. Note that the word sample and observation share the same meaning here.

The classification problem is solved by a class assignment function called the classifier. This classifier function maps from the sample space $\mathcal{Y}$ to the class space $\mathcal{C}$ and such a function can be deterministic or non-deterministic. If the classifier function is deterministic the classifier assigns one class to the sample y, whereas the non-deterministic classifier assigns to each class c a probability for the sample y to originate from that class c.

One frequently used framework for classification is to divide the sample space into regions that correspond to a class. The boundaries between these regions are called decision boundaries, these boundaries are usually a hyperplane or a combination of several hyperplanes. A lot of classifiers are based on this concept, for instance the Support Vector Machine (SVM). This geometric interpretation of the problem is illustrated in Figure 2.1, where the red line is the

boundary between the regions corresponding to the two classes.

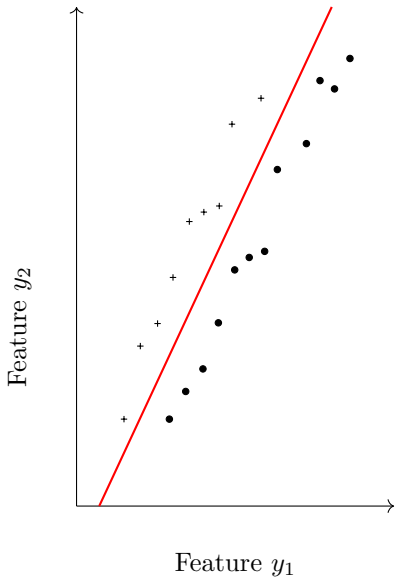


Figure 2.1: Decision boundary

But this geometric approach is only one way of looking at the classification problem. There are a lot of different approaches from a lot of different backgrounds, some of them are only applicable in a certain field whereas other approaches are more general applicable. Below we will discuss some of the most prominent methods.

- Case based methods

  The case based methods are classifying new samples by comparing the similarity with the training samples available. This similarity is measured by a metric, e.g. the Euclidean distance. The Nearest Neighbour classifier is an example of a case based classifier [9]. The classifier assigns the same class to the new sample as the class of the sample in the training set of samples which has the minimum distance to the new sample. This case based method does not need to learn, but for each new sample the algorithm has to find the sample in the training set with minimum distance to this new sample. There are a lot of variations on this approach of nearest neighbour technique.

- Logical inference

  The logical inference classifiers infer from a set of rules to which class the new sample belongs. The set of rules itself will be deduced from knowledge and will reduce the sample to a set of logical variables. From this set of variables the class is deduced. For instance a numerical feature can be reduced to a logical variable by setting the variable to true if the numerical value lies in some interval and false otherwise.

  In this framework the set of rules is usually designed by human experts discussed in [33] and [29] but can also partly be trained using the training set. In logical conjunctions [5] a certain combination of these logical variables (derived from the set of rules) is corresponding

to a class. A Decision list for each class concludes if the sample belongs to that class or not [30]. Another representation is the Decision tree, where all these different rules are combined to have one classifier. In the decision tree a rule is attached to each node and this rule will tell to which sub-tree the sample proceeds. At the end of the tree one class is associated with the sample.

The rules, the order and structure of the tree can be trained using the examples in the training set. Note that these rules in the tree can be interpreted as a division of the sample space in several regions.

- Statistical classifiers

  The previous types of classifiers assume a deterministic classifier, so each sample can belong to only one class. But if the regions in the sample space of different classes are overlapping, the sample has probabilities belonging to different classes and based on these probabilities one can decide which class is chosen. The most common choice is to choose the most probable class. Criteria for decision making will be discussed in more detail later.

  To assign a class probability to a sample we need to have a probability distribution over the sample space. This is the goal in this statistical classification approach: find the probability distribution over the sample space for a certain class. The training set is used to find these distributions. So instead of having deterministic decision boundaries the samples space is covered by a probability density function, giving us information about the corresponding class for each sample.

  The statistical methods can be parametric or non-parametric. Parametric statistics assume a certain model for the underlying probability distributions of the variables being assessed. For example one can assume the class probability distributions to be Gaussian, hence there are only two parameters to estimate for each class (mean and variance).

  Unlike the parametric statistics, non parametric statistics make no assumptions about the distributions of variables being assessed. The main difference between parametric and non parametric statistics is that the former has a fixed number of parameters, while for the latter the number of parameters grows with the size of the training set.

  The Parzen Estimator [26] is an example of the non-parametric approach where the probability function for the whole sample space is built by a linear combination of Gaussian density functions. For each sample in the training set we have Gaussian density with the mean of the density function at the training sample. So when the training set contains N training samples, the number of parameters (variance of each Gaussian and the weight of each Gaussian in the linear combination) is twice the number of samples and therefore equals $2N$.

  The Mixture Model [23] is an example of a model in between both approaches. The model is a finite sum of (parametric) probability distributions where the number of probability distributions is not increasing as the training set is increasing.

  Statistical classification theory will be discussed in more detail below.

- Artificial Neural Networks

  The Artificial Neural Networks (ANNs) as a classification mechanism is highlighted here, although it can be seen as statistical method. The ANN was studied in detail during this study. Like in the other statistical classifiers the ANN assigns probabilities to each class given some new sample. The approach for the ANN to assign these probabilities to each class is explained below.

The ANN consists of multiple layers, where each layer generally consists of three sublayers, an input layer, a hidden layer and an output layer. The input layer represents the inputs and these inputs are linearly combined by the hidden layer. Subsequently the linear combination of inputs is activated by an activation function [3] and the output of the activation function is represented in the output layer. These layers are used to build the architecture/structure of the neural network.

A simple ANN is shown in Figure 2.2. The arrows represent a layer of weights. The input layer has three features and one bias and is connected to the hidden layer by the weights. This linear combination is activated by an activation function. In classification setup this activation function is chosen such that the output is interpretable as a probability, meaning that it fulfils some of the probability properties. The outputs are a vector of probabilities over classes, hence the outputs sum up to one and lie between zero and one.

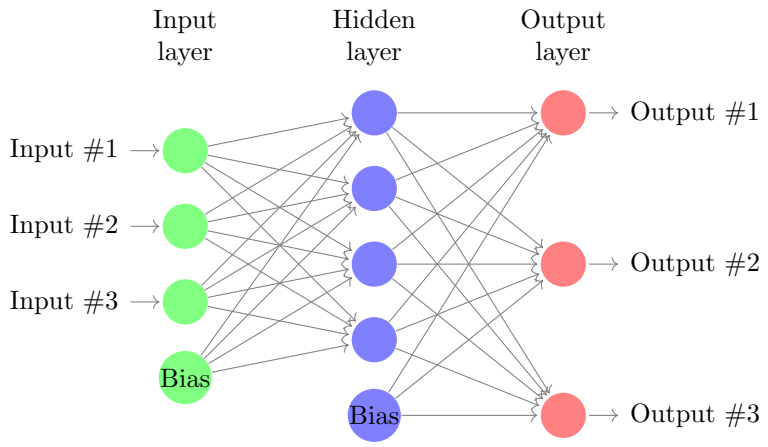


Figure 2.2: Simple graphical representation of the ANN where a vector of three inputs is classified into one of the three classes.

The ANN is learned using a labelled training set. The desired output is a vector of 1's and 0's, with a 1 for the corresponding output class of the training sample and 0's for the other outputs. Given the training set $\boldsymbol{y}$ and corresponding classes $\boldsymbol{c}$, the neural network output is given by $NN(\boldsymbol{y}, w)$, where w is the vector of all weights used in the different layers. An error function E is defined and is minimised with respect to the weights w such that the optimal weights are found,

$$\hat{w} = \arg\min_{w} E(\boldsymbol{c}, NN(\boldsymbol{y}, w))$$

This search for the optimal weights is done during the learning process. Learning is done by Error Back-Propagation [31], where the weights are initialised randomly and are updated each iteration in the direction where the error function decreases (gradient descent method), until the error function attains a (local) minimum value. Getting stuck in a local minimum is a familiar problem with gradient descent methods.

- Trade-off bias/variance

Learning from experience (training set) will always be a trade-off. In a non parametric model with a lot of parameters, like the ANN, overfitting is a general problem [4]. An over-fitted model will not perform well on new samples, hence it is important that a model shows good generalisation, i.e. the ability to classify new samples correctly. This is the trade off between bias and variance. A model which is overfitted to the training samples will have a small bias, but a high variance. Vice versa a model which is under-fitted will have a small variance but a high bias. Regularisation prevents a model to be over-fitted. Regularisation encourages smoothness of the model by penalising complexity of the model [4].

2.1 Statistical classification theory

In this section we state the statistical classification theory in more detail, as a brief introduction was given above.

Statistical classification consists in solving a classification problem via statistical inference. The underlying probability distributions and statistical methods are used to find the class probability for an observed sample, i.e. the probability of belonging to a certain class. Statistical classification theory is built upon four basic elements [2] given below.

- In statistical classification there is a underlying unknown, called *truth*, which is represented by a class c in the class space $\mathcal{C}$. The goal is to find this truth c.
- The problem of finding the truth arises due to the fact that we do not observe the truth, but we do observe *observation* y in the sample space $\mathcal{Y}$. The observations depend on the truth since the observations are often quantifying a feature or properties of the object, but measuring these features is distorted by noise, which are assumed to be random in the statistical framework. These relations are described in an observation model which models the dependency of the observation on the features of the truth. The noise assumption introduces random variables in the model and this is where statistical methods come in to help us answering questions like: what is the probability of observing y assuming that it originates from class c? This question is answered by the *likelihood function* or class conditional probability function $p_{\mathcal{Y}}(y|c)$ and arises out of the *observation model*. Using this likelihood function one concludes about the class.
- Based on these observations we have to decide which state c is the 'best' fit, this decision is made by a *decision rule* $\delta : \mathcal{Y} \to \mathcal{C}$, where δ is an allowed decision rule in the set $\mathcal{D}$. In classification theory this decision rule is called a classifier, which is a function from the sample space $\mathcal{Y}$ to the truth space $\mathcal{C}$, such a decision rule/classifier is usually an optimal choice in a optimisation problem, e.g. minimising the risk of over all decision rules in $\mathcal{D}$.
- As mentioned above the optimal classifier is optimal in the sense that it minimises an objective function, also called *loss function* $L(\delta(\cdot), c)$. In classification theory the loss function penalises a misclassification. An example of a loss function is "0/1" Loss Function, which will be discussed in section 2.2.

Thus the four basic elements are the truth/class space, the observation model, the decision rule and the loss function. As in statistics, statistical classification knows two different approaches, the classical statistics approach and the Bayesian statistics approach. Both of the statistics and their differences are discussed below.

2.1.1 Classical vs Bayesian statistics

The field of statistical decision/classification theory is divided in two different schools, the classical and the Bayesian school, hence statistical classification theory can be taught in two different ways. The statistical inference can be either the classical inference, also known as the frequentist inference, or the Bayesian inference.

One difference between both views is the modelling assumptions about the parameters of distributions. For instance, assume random variable Z normally distributed with mean μ and

variance σ^2. In the classical view these parameters are estimated and represented by a confidence interval, whereas in in the Bayesian approach these parameters are represented as random variables, containing more information than a confidence interval [18]. For example mean μ is assumed to be normally distributed with hyper parameters mean m_μ and variance s_μ.

In classical inference the accuracy of the techniques e.g. confidence intervals, unbiased estimators, are in terms of its long term repetitive accuracy. Classical statisticians consider probabilities as an objective property of nature, which can be measured accurately by sufficient repetitions of an experiment [1]. Even though in maximum likelihood estimation (MLE) this idea is not present and is also used by the classical statisticians. Classical statisticians maximise the probability of getting the data D given a certain set of hypotheses $\mathcal{H}$, e.g. model assumptions and the likelihood function $P(D|\mathcal{H})$.

While classical theory views the probability as an objective property, the Bayesian theory considers probabilities as subjective to the evidence/observations/information [10]. The term subjective refers to the fact that probabilities for identical events are different when different information is available [1]. In the Bayesian view the class is considered to be random, therefore the prior information is taken into account. The Bayesian statistician maximises the probability a certain set of hypotheses given the data,

$$P(\mathcal{H}|D) \propto P(D|\mathcal{H})P(\mathcal{H}),$$

where $P(\mathcal{H})$ is the prior probability for this set of hypothesis to be true. $P(\mathcal{H}|D)$ is called the posterior probability, after observing data D.

Having briefly introduced both schools above, we argue why the Bayesian view is adopted in the remainder of the thesis. First, the Bayesian approach takes into account a prior information about classes which plays important role in the classification problem dealt with in this thesis. Secondly the classical frequency-based interpretation of probability seems not appropriate for classification. The probability for an observation originating from a certain class does not have this frequency interpretation, since there is no sequences of outcomes for a certain class for one observation. The Bayesian classification theory is discussed in detail in the next section.

2.2 Bayesian classification theory

In this section the Bayesian classification theory is discussed using the work from Berger [1].

In Bayesian classification a priori knowledge about the truth c is used, by assuming a certain probability function $P_{\mathcal{C}}(c)$ for the state space $\mathcal{C}$. This element $P_{\mathcal{C}}(c)$ together with the basic elements, as given in section 2.1, defines the Bayesian classification problem, which can be represented by 6-tuple $(\mathcal{C}, \mathcal{Y}, \mathcal{D}, L(\delta(\cdot), c), p_{\mathcal{Y}}(y|c), P_{\mathcal{C}}(c))$. Given loss function L we want to find the optimal $\delta \in \mathcal{D}$ such that the optimality criteria holds.

Actually the notation of the probability functions can be done more rigorously. Both the prior and the likelihood function are developed under a set of modelling assumptions $\mathcal{A}$. So the notation of these probability functions should be $p_{\mathcal{Y}}(y|c, \mathcal{A}), P_{\mathcal{C}}(c|\mathcal{A}), P_{\mathcal{C}|\mathcal{Y}}(c|y, \mathcal{A})$. There are modelling assumptions in all problems we deal with and thus there are no unconditional probabilities. Although this is the correct notation, the notation including $\mathcal{A}$ will not be used, since these assumptions hold throughout the whole thesis.

Bayesian classification is named after Bayes, known for the famous Bayes' formula in (2.1). Bayes' formula is the solution in the following problem: we want to update the prior class probability $P_{\mathcal{C}}(c)$ into the posterior class probability $P_{\mathcal{C}|\mathcal{Y}}(c|y)$, i.e. the probability of class c is the truth given that we observed y. The prior probability function combined with the likelihood function $P_{\mathcal{Y}}(y|c)$ which arises out of the observation model will gives us the posterior class probability via Bayes' theorem (2.1).

We consider a discrete class space $\mathcal{C}$ with a finite number of classes and assume this space is collectively exhaustive, i.e. for all observations there is a class from which the observation originates. Since we assume class c to be a discrete random variable, we denote the prior and the posterior class probability mass functions by $P_{\mathcal{C}}(c)$, $P_{\mathcal{C}|\mathcal{Y}}(c|y)$, probability density functions of continuous random variables are denoted by p. According to Bayes' theorem the posterior class probability equals,

$$P_{\mathcal{C}|\mathcal{Y}}(c|y) = \frac{p_{\mathcal{Y}}(y|c)P_{\mathcal{C}}(c)}{p_{\mathcal{Y}}(y)}, \tag{2.1}$$

where the normalising constant is

$$p_{\mathcal{Y}}(y) = \sum_{c \in \mathcal{C}} p_{\mathcal{Y}}(y|c)P_{\mathcal{C}}(c).$$

Example

An example to indicate the influence of Bayes' formula is stated below. The example is in line with the problem of classification of UAVs/Birds.

Suppose there are two classes. There is the measurement device which is not perfect but it classifies correctly with probability 0.99. Further assume that UAVs are quite rare: one of out of 10000 flying objects is an UAV, the rest of them are birds (collectively exhaustive). What is the probability of an object being an UAV given that the measurement device classifies the object as an UAV.

Intuitively, one might think that due to the high accuracy of the measurement device this probability is large.

We want to classify an object being an UAV ($c = 1$) or a bird ($c = 0$). So we have $\mathcal{C} = \{0, 1\}$, and the outcome of the measurement device is the outcome of the sample space, UAV ($y = 1$) or bird ($y = 0$), so $\mathcal{Y} = \{0, 1\}$. The likelihood function is also given in the introduction of this problem

$$\begin{aligned} p_{\mathcal{Y}}(y = 1|c = 1) &= 0.99, \\ p_{\mathcal{Y}}(y = 0|c = 0) &= 0.99, \end{aligned}$$

and thus by the law of total probability we have that

$$\begin{aligned} p_{\mathcal{Y}}(y = 1|c = 0) &= 0.01, \\ p_{\mathcal{Y}}(y = 0|c = 1) &= 0.01. \end{aligned}$$

A priori information gives us the prior probabilities

$$\begin{aligned} P_{\mathcal{C}}(c = 1) &= 0.0001, \\ P_{\mathcal{C}}(c = 0) &= 1 - 0.0001 = 0.9999. \end{aligned}$$

All information is now available to calculate the posterior probability by (2.1)

$$\begin{aligned} P_{\mathcal{C}|\mathcal{Y}}(c = 1|y = 1) &= \frac{p_{\mathcal{Y}}(y = 1|c = 1)P_{\mathcal{C}}(c = 1)}{p_{\mathcal{Y}}(y = 1)}, \\ &= \frac{0.99 \cdot 0.0001}{0.99 \cdot 0.0001 + 0.01 \cdot 0.9999} \\ &= 0.0098, \end{aligned}$$

which is quite low, different than one might intuitively think.

A posteriori expected loss

In the Bayesian framework, the optimal classifier is based on the conditioning on the event of observing observation y, not conditional on all other possible observations which did not occur. The loss function is averaged over the state space $\mathcal{C}$ conditioned on the observation y. This is a fundamental difference between the frequentist and the Bayesian approach, the frequentist is averaging over all possible observations. So in the Bayesian approach the loss function is weighted by the posterior class probability,

$$\mathbb{E}_{\mathcal{C}|\mathcal{Y}}[L(c, \delta(y))|y] = \sum_{c \in \mathcal{C}} L(c, \delta(y)) P_{\mathcal{C}|\mathcal{Y}}(c|y).$$

Bayes' optimal classifier $\hat{\delta}(\cdot)$ is derived by minimising the a posteriori expected loss,

$$\hat{\delta}(y) = \underset{\delta(y) \in \mathcal{D}}{\arg\min} \left[\mathbb{E}_{\mathcal{C}|\mathcal{Y}} \left[L(c, \delta(y))|y \right] \right]$$

Classification under the 0/1 Loss function

A frequently used loss function is the 0/1 loss function. The 0/1 loss function penalises a misclassification with unit cost and a correct classification with zero cost,

$$L(c, \delta(y)) = \begin{cases} 1 & \text{if } c \neq \delta(y), \\ 0 & \text{if } c = \delta(y). \end{cases}$$

If we find the optimal classifier for the a posterior expected loss and with the 0/1 loss function we have that the corresponding Bayes' classifier is,

$$\begin{aligned} \hat{\delta}(y) &= \arg\min_{\delta(\cdot)\in\mathcal{D}} \left[\mathbb{E}_{\mathcal{C}|\mathcal{Y}}\left[L(c, \delta(y))|y\right]\right], \\ &= \arg\min_{\delta(\cdot)\in\mathcal{D}} \left[\sum_{c\in\mathcal{C}} L(c, \delta(y)) P_{\mathcal{C}}(c|y)\right], \\ &= \arg\min_{\delta(\cdot)\in\mathcal{D}} \left[\sum_{c\neq\delta(y)} P_{\mathcal{C}}(c|y)\right], \\ &= \arg\min_{\delta(\cdot)\in\mathcal{D}} \left[\sum_{c\in\mathcal{C}} (P_{\mathcal{C}}(c|y)) - P_{\mathcal{C}}(\delta(y)|y)\right], \\ &= \arg\min_{\delta(\cdot)\in\mathcal{D}} \left[1 - P_{\mathcal{C}}(\delta(y)|y)\right], \\ &= \arg\max_{c\in\mathcal{C}} P_{\mathcal{C}}(c|y). \end{aligned} \tag{2.2}$$

This classifier is called the maximum a posteriori (MAP) classifier. Note that the posterior class probability is proportional to $p_{\mathcal{Y}}(y|c)P_{\mathcal{C}}(c)$, therefore maximising this entity over all classes is equivalent to maximising the posterior class probability over all classes.

An example where we assume two classes $\mathcal{C} = \{0, 1\}$ and both posterior class probability are plotted in Figure 2.3.
The classifier we use is the MAP classifier, therefore for all $y \in R_1$ where, $R_1 = \{y|y > \hat{y}\}$ we have that $\delta_{MAP}(y) = 1$ and for all $y \in R_0$ where, $R_0 = \{y|y \leq \hat{y}\}$ we have that $\delta_{MAP}(y) = 0$. The total probability of misclassification is minimised by this classifier and equals,

$$\begin{aligned} P_{\min}(error) &= P(y \in R_0, c = 1) + P(y \in R_1, c = 0), \\ &= \int_{R_0} p_{\mathcal{Y}}(y|c = 1) P_{\mathcal{C}}(c = 1) dy + \int_{R_1} p_{\mathcal{Y}}(y|c = 0) P_{\mathcal{C}}(c = 0) dy. \end{aligned}$$

The optimal Bayesian classifier is actually not in line with the Bayesian view. The general Bayesian approach stops after arriving at the posterior probability density, but the optimal Bayesian classifier decides one class using the posterior density. The most information is held in the posterior densities and degrading this density to one value, e.g. MAP classifier, is a huge loss of information. In classification we are "forced" to decide a class, therefore we will use a classifier.

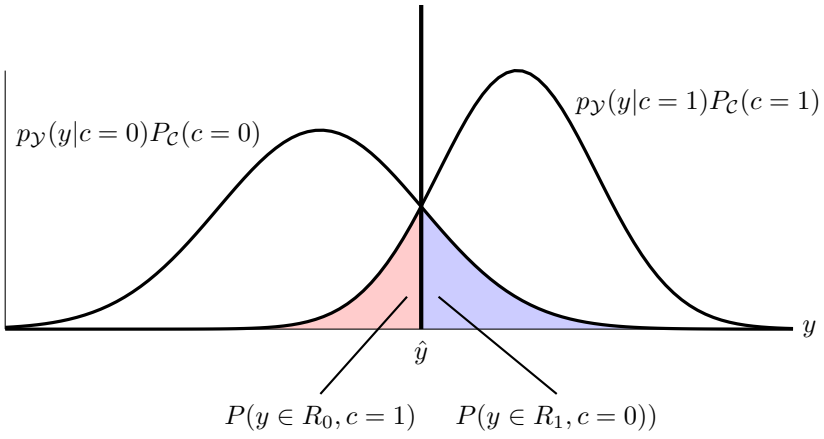


Figure 2.3: Decision boundary

2.3 Classification through hidden Markov modelling

In classification problems the truth is not directly observable, instead we observe a noisy version of features/properties of the truth, denoted by $y \in \mathcal{Y}$ [34], [7]. The features/properties (without noise), denoted by state $x \in \mathcal{X}$, depend on the class c, environment parameters and other circumstances that influence the state x. The observation model, one of the basic elements of statistical classification gives the relation between the sample/observation y and the state x. In the statistical framework the observation model introduces some random variable(s), hence we can apply the statistical /Bayesian paradigm to this modelling approach.

In many applications the features of the truth, the state x is evolving over time and therefore the model describing the state is dependent on time t. To extract information about this dynamics a sequence of observations is needed. The observation at timestep k is done at time t_k and is denoted by y_k. From K sequential observations $\{y_1, y_2, ..., y_K\}$ we obtain information about the class it is originating from. For instance when classifying if a object is moving or not, one observation about its position is not enough to conclude if the object is moving. Therefore in classification of sequential data we model the evolution of the state x_k at time t_k by a dynamical model, describing the evolution of the state. The dynamic model and an observations model can be put together in a model called the hidden Markov model, a specific type of a Bayesian Network [16].

The word observation might be ambiguous. To be clear, we will use the word observation for a set of K sequential measurements $\{y_1, y_2, ..., y_K\}$, shortly denoted by $y_{1:K}$. This one observation $y_{1:K}$ is originating from one (unknown) class. When more observations are discussed the n^{th} observation is denoted by $y^n_{1:K}$ or y^n.

A hidden Markov model is a partly observed stochastic dynamical model, it can be regarded as a Markov chain observed with noise [14]. The dynamical model for the state evolution is modelled as a hidden Markov chain denoted by $\{x_k\}_{k\geq 1}$. The general form of a dynamical model for the state x_k at time k is

$$x_k = f^c_k(x_{k-1}, w_{k-1}), \tag{2.3}$$

$$\tag{2.4}$$

where w_{k-1} is process noise. The Markov chain is hidden since the Markov chain is not observed. The stochastic process $\{y_k\}_{k\geq 1}$ is observed and depends on the Markov chain $\{x_k\}_{k\geq 1}$ via the observation model

$$y_k = g_k(x_k, v_k), \tag{2.5}$$

where v_k is the measurement noise. This dependency structure can be described in a graphical model as shown in Figure 2.4.

This dependency structure is assuming the Markov property. A stochastic process possesses the Markov property if the conditional probability distribution of the future states only depends on the present state such that for time $k > n$ we have

$$P(x_k|x_n, ..., x_1) = P(x_k|x_n).$$

So we have that the conditional probability of x_k given the past values of the states $x_{k-1}, ..., x_1$ and past values of the sequential measurements $y_{k-1}, ..., y_1$ depends only on the value of the state x_{k-1}. The conditional probability of y_k given the past values of the states $x_k, ..., x_0$ and past values of the sequential measurements $y_{k-1}, ..., y_1$ depends only on the past value of the state

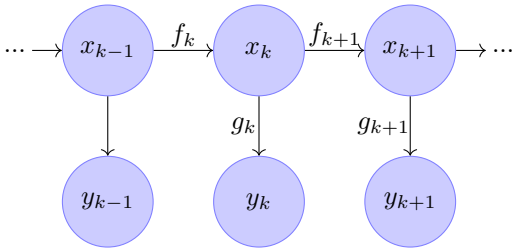


Figure 2.4: The graphical representation of a hidden Markov model, where the arrows represent dependencies between state x_k and observations y_k.

x_k.

The hidden Markov chain is a stochastic process originating from a certain class, since the function f^c in the dynamic model is class dependent. Given observation $y_{1:K}$ a class is inferred by the optimal Bayes' classification rule also known as the MAP classifier (2.2) [16]. The posterior class probability for class c equals

$$P_{\mathcal{C}}(c|y_{1:K}) = \frac{p_{\mathcal{Y}|\mathcal{C}}(y_{1:K}|c)P_{\mathcal{C}}(c)}{\sum_{c=1}^{M} p_{\mathcal{Y}|\mathcal{C}}(y_{1:K}|c)P_{\mathcal{C}}(c)}. \tag{2.6}$$

The likelihood function $p_{\mathcal{Y}|\mathcal{C}}(y_{1:K}|c)$ is the entity of interest in equation (2.6). Applying the definition of the conditional probability gives,

$$p_{\mathcal{Y}|\mathcal{C}}(y_{1:K}|c) = p(y_K|y_{1:(K-1)}, c)p(y_{1:(K-1)}|c),$$

applying the conditional probability definition $K-1$ times gives,

$$\begin{aligned} p(y_{1:K}|c) &= \prod_{k=1}^{K} p(y_k|y_{1:(k-1)}, c), \\ &= \prod_{k=1}^{K} \int_{\mathcal{X}} p(y_k|x_k, c)p(x_k|y_{1:k-1}, c)dx_k, \end{aligned}$$

using the Chapman-Kolmogorov equation we have

$$p(x_k|y_{1:k-1}, c) = \int_{\mathcal{X}} p(x_k|x_{k-1}, c)p(x_{k-1}|y_{1:k-1}, c)dx_{k-1},$$

The density $p(y_k|x_k, c)$ arises from the observation model (2.5) while the density $p(x_k|x_{k-1}, c)$ arises from the dynamic model (2.3). The density $p(x_{k-1}|y_{1:k-1}, c)$ depends on both models and is not always tractable. There is an analytical expression for the linear Gaussian models which can be computed as a recursive update.

The framework for classification is now complete. In the next section the learning process of parameters is discussed.

2.4 Learning parameters

A model is a representation of the 'world', but a perfect match between world and model cannot be attained because there are too many uncertainties in the measured 'world', data. In the classification problem the main goal is to find the corresponding class for an observation. The dynamical models and the observation models have a lot of parameters and not all parameters are known. The unknown parameters need to be learned or estimated. In essence learning and estimation are the same, but we use the words to indicate different parameters. The parameters that have to be learned are parameters that have a common value for a certain class or all objects. One might think of the variance of the process noise for a certain class. These parameters can be learned from a set of observations. Next to the parameters that need to be learned, there are parameters that need to be estimated. The estimation of these parameters is done based on one observation since the parameter value can be different for every observation.

The set of observations can also be used to learn the structure of models (hidden Markov models, dynamic model) or learn hyper-parameters, which are parameters of the distribution of model parameters.

To learn we need experience. The experience comes from observations and in classification theory we have unlabelled observations and labelled observations. For an unlabelled observation y the class it is originating from is unknown, whereas for a labelled observation the class is known. For labelled observations the estimation of the parameters is 'easier', since the class is already known and therefore there is less uncertainty in play. A labelled set of observations can be used to learn general model parameters or their distributions (parameters), but also can learn specific class parameters. The unknown model parameters in the model for class c are denoted by χ_c and χ contains all unknown model parameters of all classes. When learning with labelled observations the process is called supervised learning. Although supervised learning is more accurate and a better approach because there is less uncertainty about the observations, the cost of acquiring the supervised observations are high. The high costs arise due to the labelling of the observations, which has to be done by humans. For instance in the case of UAVs, a human must identify that a UAV is flying in the sight of the radar and thus the observations that are acquired can be labelled with the UAV class. We assume that we don't have labelled observations, so we will focus on unsupervised learning with non-labelled observations. These unsupervised learning techniques are regarded to be more general and can easily be adapted to the supervised learning with labelled observations.

The prior class probabilities denoted by $P(c)$ may also be assumed unknown. In the case of labelled observations one can estimate these prior probabilities using the number of observations for each class.

We state below the technique called Maximum Likelihood Adaptive Neural Systems (MLANS) for simultaneous classification and learning of model parameters is stated [27]. It is an off-line learning technique and it learns from a static dataset, i.e. all observations are presented simultaneously [21]. The technique is quite similar to the Expectation-Maximisation algorithm. There is a slight difference in the objective function both techniques try to optimise.

2.4.1 MLANS algorithm

The Maximum Likelihood Adaptive Neural System is designed for a set of unlabelled observations, which is used to learn the parameters and to classify all observations [27]. The model parameters are learned, such that the similarity between the feature models (in this thesis the hidden Markov models) and observations is maximised. The similarity measure is the likelihood function. This likelihood function is a function of the model parameters we want to learn and the probability density function for the observations given the model parameters.

Let us specify this likelihood function $l(\chi|y^i)$ for an unlabelled observation y^i,

$$\begin{aligned} l(\chi|y^i) :&= p(y^i|\chi), \\ &= \sum_{c\in C} p(y^i, c|\chi), \\ &= \sum_{c\in C} p(y^i|c,\chi)P(c). \end{aligned} \tag{2.7}$$

For a set of N unlabelled independent observations $\boldsymbol{y} = (y^1, y^2, ..., y^N)$ the total likelihood l equals,

$$\begin{aligned} l(\chi|\boldsymbol{y}) &= p(\boldsymbol{y}|\chi), \\ &= \prod_{i=1}^{N} p(y^i|\chi), \\ &= \prod_{i=1}^{N} \sum_{c\in C} p(y^i|c,\chi)P(c). \end{aligned}$$

The log-likelihood is used in this thesis, since the log-likelihood is more convenient to work with for numerical reasons. Also since the log-likelihood function is monotonically increasing, it attains its maximum value at the same points as the likelihood function.

$$\begin{aligned} ll(\chi|\boldsymbol{y}) :&= \log\left(l(\chi|\boldsymbol{y})\right), \\ &= \log\left(\prod_{i=1}^{N} \sum_{c\in C} p(y^i|c,\chi)P(c)\right), \\ &= \sum_{i=1}^{N} \log\left(\sum_{c\in C} p(y^i|c,\chi)P(c)\right). \end{aligned}$$

Maximisation of the log-likelihood is achieved by differentiating with respect to parameters χ. In addition, if the prior probabilities $P(c)$ are assumed to be unknown and need to be estimated, the prior class probabilities are required to sum up to one,

$$\sum_{c\in C} P(c) = 1. \tag{2.8}$$

To account for this constraint a Lagrange multiplier is added, such that the semi log-likelihood

function becomes

$$ll'(\chi|\boldsymbol{y}) := ll(\chi|\boldsymbol{y}) + \mu\left(\sum_{c\in C} P(c) - 1\right),$$
$$= \sum_{i=1}^{N} \log\left(\sum_{c\in C} p(y^i|c,\chi)P(c)\right) + \mu\left(\sum_{c\in C} P(c) - 1\right). \tag{2.9}$$

The semi-log likelihood gradient with respect to the model parameters $\chi_{\tilde{c}}$ for class $\tilde{c}$ is given by

$$\frac{\partial ll'(\chi|\boldsymbol{y})}{\partial \chi_{\tilde{c}}} = \sum_{i=1}^{N} P(\tilde{c}|y^i,\chi)\frac{\partial}{\partial \chi_{\tilde{c}}}\log\left(p(y^i|\tilde{c},\chi)P(\tilde{c})\right), \tag{2.10}$$

where $\frac{\partial}{\partial \chi_{\tilde{c}}}\log\left(p(y^i|\tilde{c},\chi)P(\tilde{c})\right)$ depends on the model assumptions for class $\tilde{c}$. The derivation of this gradient can be found in Appendix B.1.

The semi log-likelihood in (2.9) is also maximised with respect to the prior class probabilities and the derivation can be found in Appendix B.1,

$$\frac{\partial ll'(\chi|\boldsymbol{y})}{\partial P(\tilde{c})} = \sum_{i=1}^{N} \frac{P(\tilde{c}|y^i,\chi)}{P(\tilde{c})} + \mu,$$

Equating to zero gives

$$P(\tilde{c}) = -\sum_{i=1}^{N} \frac{P(\tilde{c}|y^i,\chi)}{\mu},$$

and satisfying constraint (2.8) gives $\mu = -N$ and thus

$$P(\tilde{c}) = \sum_{i=1}^{N} \frac{P(\tilde{c}|y^i,\chi)}{N}, \tag{2.11}$$

where the posterior probability is

$$P(\tilde{c}|y^i,\chi) = \frac{p(y^i|\tilde{c},\chi)P(\tilde{c})}{\sum_{c\in C} p(y^i|c,\chi)P(c)}. \tag{2.12}$$

Since the parameters, which may include the prior probabilities, are unknown an two step iterative scheme is employed. Starting with initial parameter values the posterior probabilities (2.12) are computed. Next the prior probabilities and other parameters are updated using (2.11) and (2.10) respectively. The updating of the parameters is done by the gradient ascent method or directly if an analytical optimal solution is tractable. The pseudo code of this scheme is written in Algorithm 1. The method is proven to converge to a local maximum, [28].

2.4.2 Expectation Maximisation algorithm

The Expectation-Maximisation (EM) algorithm is another iterative method to compute maximum likelihood estimates for a set of parameters if the observations are unlabelled. Whereas the

Data: dataset of independent unlabelled observations y^i
initialisation;
initialise $\hat{\chi}^0$ and $ll'(\hat{\chi}^0)$;
it=0;
while $|ll'(\chi^{it+1}) - ll'(\chi^{it})| < \epsilon$ **do**
  $ll'(\chi|\hat{\chi}^{it}) \leftarrow P_{\mathcal{C}|\mathcal{Y}}(c|y, \chi^{it})$;
  $\hat{\chi}^{it+1} \leftarrow \arg\max_{\chi} ll'(\chi|\hat{\chi}^{it})$;
  $it = it + 1$;
end
return $\hat{\chi}^{it}$;

Algorithm 1: The MLANS algorithm

MLANS algorithm optimises for the sum of class log likelihoods, the EM algorithm optimises the a posterior expected log-likelihood [12],

$$\mathbb{E}_{C|Y,\hat{\chi}}\left[ll(\chi|\boldsymbol{y}, C)\right] = \sum_{c\in\mathcal{C}} ll(\chi|\boldsymbol{y}, c)P(c|y^i, \hat{\chi}), \tag{2.13}$$

The EM algorithm consists of two steps, the Expectation step and the Maximisation step and aims to find the true values for the set of parameters. In the expectation step (2.13) the posterior probabilities are calculated using the current estimates for the parameters. This expectation is then maximised with respect to the set parameters in the Maximisation step, giving us a new estimate for the set of parameters. Both steps are iterated and this iterative method is shown to converge to a local maximum for a family of exponential densities, including the Gaussian density [37].

The two steps in the EM algorithm are iterated till the difference between expected values in two successive iteration steps is smaller than threshold ϵ. The two steps at each iteration are,

$$\begin{aligned} Q(\chi|\hat{\chi}^{it}) &:= \mathbb{E}_{C|Y,\hat{\chi}^{it}}\left[ll(\chi|\boldsymbol{y}, C)\right], \\ \hat{\chi}^{it+1} &= \arg\max_{\chi} Q(\chi|\hat{\chi}^{it}), \end{aligned} \tag{2.14}$$

where it is the number of iterations and $Q(\chi|\hat{\chi})$ denotes the expected value of the log likelihood given y. The gradient for the EM algorithm is given by,

$$\begin{aligned} \mathbb{E}_{C|Y,\hat{\chi}}\left[ll(\chi|\boldsymbol{y}, C)\right] &= \sum_{i=1}^{N} \mathbb{E}_{C|Y,\hat{\chi}}\left[\log(p(y^n, C|\chi))\right], \\ &= \sum_{i=1}^{N}\sum_{c\in\mathcal{C}} \left[\log(p(y^n, c|\chi))P(c|y^i, \hat{\chi})\right], \end{aligned}$$

and thus we have,

$$\frac{\partial\ \mathbb{E}_{C|Y,\hat{\chi}}\left[ll(\chi|\boldsymbol{y}, C)\right]}{\partial\chi_{\tilde{c}}} = \sum_{i=1}^{N}\left[P(\tilde{c}|y^i, \hat{\chi})\frac{\partial\log(p(y^n, \tilde{c}|\chi))}{\partial\chi_{\tilde{c}}}\right]. \tag{2.15}$$

Maximising (2.9) and (2.14) results in equal optimal parameters since the gradients in the MLANS algorithm (2.10) and EM algorithm (2.15) are actually equivalent. The prior prob-

ability update is also equivalent. The gradient (2.15) with respect to $P(\tilde{c})$ is

$$\begin{aligned}\frac{\partial\ \mathbb{E}_{C|Y,\hat{\chi}}\left[ll(\chi|\boldsymbol{y},C)\right]}{\partial P(\tilde{c})} &= \sum_{i=1}^{N}\left[P(\tilde{c}|y^i,\hat{\chi})\frac{\partial \log(p(y^n,\tilde{c}|\chi))}{\partial P(\tilde{c})}\right] \\ &= \sum_{i=1}^{N}\left[P(\tilde{c}|y^i,\hat{\chi})\frac{1}{P(\tilde{c})}\right]. \end{aligned} \tag{2.16}$$

Gradient (2.16) with constraint (2.8) gives equation (2.11), hence we will find the same solution given the same initial estimates for the parameters and the prior probabilities.

Data: dataset of independent unlabelled observations y^i
initialisation;
initialise $\hat{\chi}^0$ and priors $P^0(c)$;
$it = 0$;
while $|Q(\hat{\chi}^{it}|\hat{\chi}^{it}) - Q(\hat{\chi}^{it-1}|\hat{\chi}^{it-1})| < \epsilon$ **do**
  $Q(\chi|\hat{\chi}^{it}) \leftarrow P_{\mathcal{C}|\mathcal{Y}}(c|y,\hat{\chi}^{it})$ (E-step) ;
  $\hat{\chi}^{it+1} \leftarrow \arg\max_{\chi} Q(\chi|\hat{\chi}^{it})$ (M-step) ;
  $it = it + 1$;
end
return χ^{it} ;

Algorithm 2: The EM algorithm

The intuition of this learning is as follows. The parameter is estimated for each observation and is then weighted by the posterior probability of that observations. By this concept the observations with a high certainty, i.e. high posterior probability for one of the classes are weighted more, since these observations are likely to give us a good estimate for the parameter for that class.

2.4.3 Maximum likelihood estimation

For the model parameters that differ for each observation maximum likelihood estimation is used, but now these estimates solely depend on one observation. These parameters χ^c in the c class model are estimated by the maximum likelihood estimate for the particular class. For the maximum likelihood estimator we compute,

$$\begin{aligned}\hat{\chi}^{\tilde{c}} &= \arg\max_{\chi^{\tilde{c}}} p(y_{1:K}|\chi), \\ &= \arg\max_{\chi^{\tilde{c}}} \sum_{c\in\mathcal{C}} p(y_{1:K},c|\chi), \\ &= \arg\max_{\chi^{\tilde{c}}} p(y_{1:K},\tilde{c}|\chi).\end{aligned}$$

Although the estimation seems straightforward, the estimation can be costly due to the dimension of the parameter space since the computation time increases exponentially with the parameter space dimension, this phenomena is called the curse of dimensionality. Estimating d parameters each having a partition of M points, the computation of the maximum likelihood needs to be done M^d times.

Chapter 3

Classification of UAVs and birds

In this chapter the main contribution of this thesis is stated. First we model the UAV and birds by a single point scatterer. We build up the dynamical class model and corresponding observation model for the received radar signal of a UAV and bird. This developed hidden Markov model is used in the Bayesian statistical framework to classify and simultaneously learn model parameters. Secondly the UAV and birds are modelled by multiple point scatterer models upon which the classifier is based.

In Chapter 2 we have presented background knowledge that will be used to build the classifier. First we justify and explain the approach we use to build the classifier. Next the UAV and bird signal models are developed using the background knowledge on radars. Subsequently we develop corresponding hidden Markov models and finally present the classifier.

3.1 Classification approach

The main goal of this project is to construct a classifier to differentiate between UAVs and birds and this classifier should be robust to different radar parameter values and environment parameter values.

From the different approaches to build a classifier, we choose to build a statistical classifier. We model the noises in the signal models as random variables, therefore a statistical approach is more appropriate.

The statistical classifier can be either based on a parametric model or a non-parametric model. The non-parametric approach doesn't exploit any prior knowledge. It assumes no a priori specific structure, but a very general structure and the number of parameters is increasing if more observations are available. An example of the non-parametric approach is the Parzen Estimator [26]. To learn a specific model from the general structure the non-parametric approach needs a lot of observations. If no prior knowledge is available about the phenomena which is modelled, this non-parametric approach is a powerful technique and can model the phenomena well, since a lot of these techniques can approximate a large number of functions up to arbitrary accuracy. For example, the ANN can approximate the continuous functions on compact subsets of $\mathbb{R}^n$ with arbitrary precision [11], hence the ANN is frequently used in classification problems where the underlying model is not known. So the non-parametric approach is a powerful tool, but it has its limitations. The general structure of these non-parametric classifiers is trained on a specific set of labelled observations. For new observations lying outside of the domain in the sample space $\mathcal{Y}$ which is covered by the training data extrapolation is needed. The non-parametric techniques have to extrapolate, but when no knowledge is available about the underlying model in the new domain this extrapolation is poor [20]. So these techniques are expected to perform poor in extrapolation, since no observations are available in these domains we want to extrapolate over.

The parametric approach is depending on a finite and smaller number of parameters and is built upon prior knowledge which leads to a specific model structure. Since models are never perfect, data is used to make the model fits reality better.

In our approach to tackle the problem we take the parametric approach, two arguments will support the choice for this approach.

Firstly, in our problem the radar parameters and environment parameters can take an infinite number of distinct values, since these are continuous parameters, e.g. carrier frequency, aspect angles, hence the non-parametric approach needs infinite amount of labelled observations to learn a model. Labelled observations are scarce, specially the observations on UAVs.

Secondly, this phenomena of radar waves reflecting on a target is a physical phenomena and therefore can be described by physical laws so there is a lot of a priori knowledge, hence a proper underlying model with the radar parameters and environment parameter arises naturally.

Though the performance of this approach is liable to the correctness of the structure of model and model parameters, this approach has still some flexibility in it. The model structure can be adapted and model parameters can be learned from observations. Adaptive learning is an important aspect of the approach, because it will make the approach more robust to observations that are not in line with the models.

In section 2.3 it was shown how to compute the classifier based on a hidden Markov model. The likelihood $p(y_{1:K}|c)$ depends on the product of $p(y_k|y_{1:k-1}, c)$ terms. These conditional densities are analytically tractable and therefore the likelihood is tractable, as given in Appendix B.2, but likelihood gradients are harder to work with since they become analytically complex.

Therefore we use the approximation

$$p(y_k|y_{1:k-1}, c) \approx p(y_k|y_{k-1}, c),$$

such that the likelihood is approximately,

$$p(y_{1:K}|c) \approx \prod_{k=1}^{K} p(y_k|y_{k-1}, c),$$

for which the gradient can be calculated exactly.

As we shall see below our hidden Markov model for class c of the form

$$\begin{aligned} x_{k+1} &= F_k^c x_k + w_k, \\ y_{k+1} &= x_{k+1} + v_{k+1}. \end{aligned} \tag{3.1}$$

where v_i, w_i are assumed to be independent zero mean circular complex Gaussian noise with variance C_v and C_w respectively for $i = 1, .., K$. Complex circular complex Gaussian noise is explained in Appendix B.3.

Under these assumptions, the approximating conditional density is given by

$$p(y_k|y_{k-1}, c) = \mathcal{CN}(y_k \Big| F_{k-1}^c y_{k-1}, |F_{k-1}^c|^2 C_v + C_w + C_v)$$

and the approximation log likelihood is given by,

$$\begin{aligned} ll = &\sum_{k=1}^{K} \ln\left(\frac{1}{\pi(|F_{k-1}^c|^2 C_v + C_v + C_w)}\right) \\ &+ \sum_{k=1}^{K}\left[\frac{-|y_k|^2 - |y_{k-1}|^2 + 2Re\left(y_k \overline{F_{k-1}^c y_{k-1}}\right)}{(|F_{k-1}^c|^2 C_v + C_v + C_w)}\right], \end{aligned} \tag{3.2}$$

where $Re(z)$ gives the real part of z and $\overline{z}$ gives back the complex conjugate of z.

Throughout the rest of this thesis this approximating log likelihood is used.

3.2 Radar return signal

Before the UAV/bird signal models are developed, the common modelling part for both UAVs and birds is done. We will start with a model for the radio waves the radar is transmitting.

A radar transmits a radio frequency electro magnetic signal s_T which can be modelled as a complex sinusoid [6]

$$s_T(t) = A(t)e^{-j[2\pi f_d t + \Phi_0]}, \tag{3.3}$$

where $A(t)$ is a time varying amplitude, initial phase Φ_0, carrier frequency f_d and imaginary unit j.

We assume that the receiver is located in the same place as the transmitter. The received signal s_R travels twice the distance $R(t)$ at time t between the transmitter/receiver and the point scatterer and is dependent on the reflectivity ρ of the scatterer, The transmitted signal is reflected by a point scatterer P and the reflected signal is received by the radar [6]

$$s_R(t) = \rho A(t)e^{-j[2\pi f_d (t-\tau) + \Phi_0]}, \tag{3.4}$$

where it takes time τ to travel back and forth to the scatterer. First we discuss some simplifying assumptions and argue why they are reasonable.

A.1 The source and the receiver of the radio signal share the same location.

Such a radar is called a monostatic radar [8] and it is a conventional configuration for a radar.

A.2 The point scatterer is in the far field.

This assumption is a common one in the field of radar modelling [19]. Since the target is in the far field we can approximate the incident wave and the reflected wave by a plane wave, hence the radar can be represented by a point source and receiver, which simplifies the model significantly. The far field region is dependent on the size of the radar and the carrier frequency.

A.3 $c \gg v$ where c is the speed of light and v the speed of the point scatterer.

This assumption implies that the distance to the point scatterer at the moment of reflection can be approximated by the distance to the point scatterer $R(t)$ at receiving time t and thus we have

$$\tau = \frac{2R(t)}{c}, \tag{3.5}$$

therefore the received signal (3.4) can be written explicitly. The assumption is realistic and if this assumption is not made the model would be implicitly defined, which makes it a lot harder to work with.

A.4 The translation velocity of the target is zero.

This assumption is purely made for simplification. The distance model described below could easily be extended with a translational velocity., but extra parameters come in and this increases the parameters space exponentially, leading to parameter estimation problems, e.g. computation time, hence we made this assumption.

A.5 The amplitude $A(t)$ and reflectivity ρ are assumed te be non random and constant, $A(t) = A, \ \forall t$.

First we can assume $A(t)$ to be constant, since this is a sensor parameter which can be set. The reflectivity parameter ρ is in general not known and can vary over time. But for the sake of simplicity we make this assumption.

A.6 The length of the blade or bird wing is small compared to the range of the centre of rotation.

To get a simplified expression for the distance of the point we need this assumption which is a reasonable assumption to make since the length of a wing or blade of a mini UAV will not be larger than one meter and the range will most likely be larger than at least 10 meters.

These are the basic assumptions throughout the rest of the thesis. More assumptions will be made throughout the extension of the model. Now we derive the model for the distance $R_P(t)$ to the single point scatterer P.

Given a coordinate frame with the center at the source/receiver of the radar. Under assumptions A.1-A.6 we have that point scatterer P is rotating around a fixed center O at range R_0 with azimuth angle α and elevation angle β. The Cartesian coordinates (x_O, y_O, z_O) of the point O in the radar coordinate frame are then related as

$$\begin{aligned} x_O &= R_0 \cos(\beta)\cos(\alpha), \\ y_O &= R_0 \cos(\beta)\sin(\alpha), \\ z_O &= R_0 \sin(\beta). \end{aligned} \tag{3.6}$$

The coordinate frame (x, y, z) are illustrated in Figure 3.1.

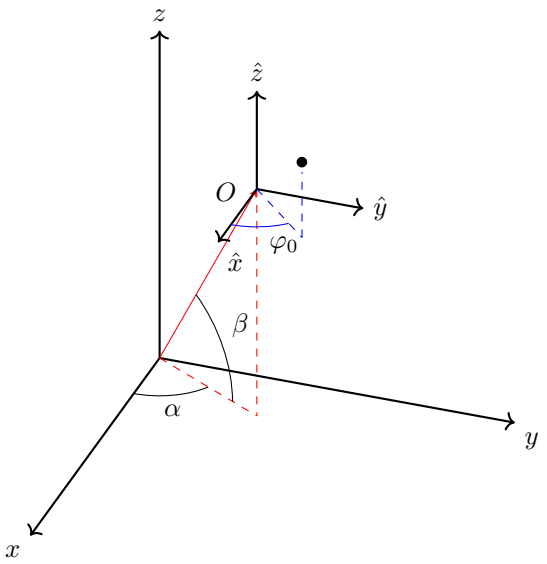


Figure 3.1: The fixed center O is placed at range R_0 and with azimuth angle α and elevation angle β in the radar coordinate system. There is a local coordinate frame $(\hat{x}, \hat{y}, \hat{z})$ with center O in which the point scatterer P is expressed with respect to origin O.

3.3 Single point scatterer model for a UAV

In this section we exploit prior knowledge so that we can model the reflected radar signal for a Unmanned Aerial Vehicle (UAV). We start by modelling one rotating point scatterer as a representation of a point of a blade of the UAV.

The location of point P is expressed in the local coordinate frame with origin O and without loss of generality we can assume that this local coordinate frame has the same orientation as the radar coordinate frame. The Cartesian coordinates (x_P, y_P, z_P) of the rotating point P in the local coordinate frame at time t are

$$\begin{aligned} x_P(t) &= l_P \cos(\varphi_0 + \omega_c t), \\ y_P(t) &= l_P \sin(\varphi_0 + \omega_c t), \\ z_P(t) &= z_0, \end{aligned}$$

with initial azimuth angle φ_0 , distance l_P between $\hat{z}$-axis and point P , rotation (rotor) speed ω_c and the fixed height of the point is z_0.

The distance between the point scatterer P and the radar source as a function of time t is,

$$\begin{aligned} R_P(t) &= \sqrt{(x_O + x_P)^2 + (y_O + y_P)^2 + (z_O + z_P)^2} \\ &= \left[R_0^2 + l_P^2 + 2R_0 l_P \cos(\beta)\cos(\alpha - (\varphi_0 + \omega_c t)) + 2R_0 z_0 \sin(\beta)\right]^{1/2} \qquad (3.7) \\ &\approx R_0 + l_P \cos(\beta)\cos(\alpha - \omega_c t - \varphi_0) + z_0 \sin(\beta). \qquad (3.8) \end{aligned}$$

A justification of the approximation step going from equation (3.7) to equation (3.8) is given in Appendix B.4.

Next we derive the expression for the received signal (3.4) given the distance expression (3.8).

The received signal at time t depends on the signal sent from the source at time $t - \tau$ and the distance at time t by assumption A.3. The distance expression (3.8) combined with equation (3.5) are substituted into the returned signal (3.4). The received signal is now a function of the class dependent variables: rotation speed ω_c and radius l_P, and as a function of radar/sensor variables: initial phase Φ_0, amplitude A, angular carrier frequency $\omega_d = 2\pi f_d$ and wavelength $\lambda = \frac{c}{f_d}$

$$s_{R_P}(t) = \rho A e^{-j[\omega_d t + \Phi_0]} e^{j\frac{4\pi}{\lambda}[R_0 + z_0 \sin(\beta)]} e^{\frac{j4\pi}{\lambda} l_P \cos(\beta)\cos(\alpha - \omega_c t - \varphi_0)}. \qquad (3.9)$$

Since the observations are samples we discretise the signal. At time t_k the received signal is

$$\begin{aligned} s_k :&= s_R(t_k) \\ &= \rho A(t) e^{-j[\omega_d t_k + \Phi_0]} e^{j\frac{4\pi}{\lambda}[R_0 + z_0 \sin(\beta)]} e^{\frac{j4\pi}{\lambda} l_P \cos(\beta)\cos(\alpha - \omega_c t_k - \varphi_0)} \\ &= \tilde{A} e^{-jB} e^{-j[\omega_d t_k]} e^{jG\cos(\alpha - \omega_c t_k - \varphi_0)}, \end{aligned}$$

where for notational convenience we have introduced the constants

$$\begin{aligned} \tilde{A} &= \rho A, \\ B &= \Phi_0 - \frac{4\pi}{\lambda}\left[R_0 + z_0 \sin(\beta)\right], \\ G &= \frac{4\pi}{\lambda} l_P \cos(\beta). \end{aligned}$$

At time $t_{k+1} = t_k + \Delta t$, where Δt is the sampling time of the receiver, the state s_{k+1} can be written as a function of the previous state s_k

$$s_{k+1} = e^{j\Phi_k^c} s_k,$$

where

$$\Phi_k^c = -\omega_d \Delta t + G\cos(\alpha - \omega_c t_{k+1} - \varphi_0) - G\cos(\alpha - \omega_c t_k - \varphi_0)).$$

Next we add to this evolution of state x_k the process noise which is zero mean circular complex Gaussian noise. The process noise makes the dynamic model more robust to possible unmodelled dynamics.

For $k = 1$

$$x_1 = s_1 + w_1,$$

and for $k > 1$,

$$x_{k+1} = e^{j\Phi_k^c} x_k + w_k. \tag{3.10}$$

Together with the observation model (3.1) this gives a hidden Markov model. This hidden Markov model is dependent on the class parameters, so the underlying class models are hidden Markov models. The class dependent parameters for the single point scatterer model are defined as the rotation speed ω_c and the radius l_P.

The approximation of the log likelihoods can now be computed by (3.2) and subsequently the approximation of the class posterior probabilities.

3.4 Multiple point scatterers model for a UAV

In the previous section we assumed that the UAV can be modelled as a single point scatterer. In this section the UAV is modelled by multiple point scatterers, a line of scatterers. The signal model for a rotating blade is stated in[32].

The blade model is evaluated under the same assumptions as before, so the blade will rotate around the $\hat{z}$ axis at height z_0. The one end of the blade is located L_1 unit away from the $\hat{z}$-axis and has length L. The point scatterer signal model (3.9) is used to derive the blade model. Further we make the assumption,

A.7 The blade can be modelled as a continuous line of point scatterers. It is a homogeneous linear rigid antenna [22].

This assumption makes the model less complex, i.e. fewer parameters are used. Further, it is reasonable to assume that a blade is made of homogeneous material such that all point scatterers have the same reflectivity ρ.

The model for a blade arises when the model for a single point scatterer is integrated over a line starting at L_1 to the tip of the blade located at $L_2 = L_1 + L$,

$$\begin{aligned} s_{\text{blade}}(t) &= \int_{L_1}^{L_2} s_{R_P}(l_P) dl_P, \\ &= \rho A e^{-j[\omega_d t + \Phi_0]} e^{j\frac{4\pi}{\lambda}[R_0 + z_0 \sin(\beta)]} \cdot L e^{\frac{j4\pi}{\lambda}\frac{L_1+L_2}{2}\cos(\beta)\cos(\alpha-\omega_c t-\varphi_0)} \\ &\quad \cdot \operatorname{sinc}\left(\frac{4\pi}{\lambda}\frac{L}{2}\cos(\beta)\cos(\alpha-\omega_c t-\varphi_0)\right), \end{aligned}$$

where the sinc function is defined as,

$$\operatorname{sinc}(x) = \frac{\sin(x)}{x}.$$

Notice that the power of the reflected signal $|s_{\text{blade}}(t)|^2$ attains it maximum when the orientation of the blade is perpendicular to the direction of the incident wave. In fact the 'sinc' term equals one if the blade is perpendicular to the incident wave.

The blade model can easily be extended to a rotor model as done in [22] by making the following assumption,

A.8 The rotor has N_b blades and these blades are uniformly separated in angle, e.g. for a rotor with 3 blades the angle between two successive blades is $\frac{2\pi}{3}$.

This assumption is realistic since all rotors have the blades separated by a uniform angle, but the number of blades attached to the rotor can differ. The most common number of blades are two, three or four blades per rotor.

A.9 No shielding occurs, meaning that we assume that all blades are visible for the radar at all times.

Although shielding can occur at some orientations of the rotor with respect to the radar, the assumption leads to a good theoretical model.

So the initial angle of the n^{th} blade is $\varphi_n = \varphi_0 + \frac{2\pi(n-1)}{N_b}$. The signal of the multiple blades can be modelled as the sum of the blade signals and therefore we have the rotor signal,

$$s_{\text{rotor}}(t) = L\rho A e^{-j[\omega_d t+\Phi_0]} e^{j\frac{4\pi}{\lambda}[R_0+z_0\sin(\beta)]} \cdot \sum_{n=1}^{N_b} e^{\frac{j4\pi}{\lambda}\frac{L_1+L_2}{2}\cos(\beta)\cos(\alpha-\omega_c t-\varphi_0-\frac{2\pi(n-1)}{N_b})}$$
$$\cdot \operatorname{sinc}\left(\frac{4\pi}{\lambda}\frac{L}{2}\cos(\beta)\cos(\alpha-\omega_c t-\varphi_0-\frac{2\pi(n-1)}{N_b})\right). \tag{3.11}$$

Now we derive the dynamical model for the signal model (3.11).

At time t_k the received signal is of the form,

$$\begin{aligned} s_k :&= s_{\text{rotor}}(t_k) \\ &= L\tilde{A}e^{-jB}e^{-j[\omega_d t_k]} \sum_{n=1}^{N_b} e^{j\frac{L_1+L_2}{2}M\cos(\alpha-\omega_c t_k-\varphi_0-\frac{2\pi(n-1)}{N_b})} \\ &\quad \cdot \operatorname{sinc}\left(M\frac{L}{2}\cos(\alpha-\omega_c t_k-\varphi_0-\frac{2\pi(n-1)}{N_b})\right), \end{aligned}$$

where for notational convenience we have introduced the constants

$$\begin{aligned} \tilde{A} &= \rho A, \\ B &= \Phi_0 - \frac{4\pi}{\lambda}\left[R_0 + z_0\sin(\beta)\right], \\ M &= \frac{4\pi}{\lambda}\cos(\beta). \end{aligned}$$

At time $t_{k+1} = t_k + \Delta t$, where Δt is the sampling time of the receiver, the state s_{k+1} can be written as a function of the previous state s_k if $|s_k| > 0$,

$$s_{k+1} = \Gamma_k^c s_k,$$

where

$$\Gamma_k^c = e^{-j[\omega_d(\Delta t)]}\frac{\sum_{n=1}^{N_b} e^{j\frac{L_1+L_2}{2}M\Lambda_k^c}\operatorname{sinc}\left(M\frac{L}{2}\Lambda_k^c\right)}{\sum_{n=1}^{N_b} e^{j\frac{L_1+L_2}{2}M\Lambda_k^c}\operatorname{sinc}\left(M\frac{L}{2}\Lambda_k^c\right)}, \tag{3.12}$$

where

$$\Lambda_k^c = \cos(\alpha-\omega_c t_k-\varphi_0-\frac{2\pi(n-1)}{N_b}).$$

If $|s_k| = 0$ we can simply add a small positive number $\epsilon > 0$ to the denominator in (3.12) to prevent dividing by zero.

Similarly to the single point scatterer model we assume additive zero mean circular complex Gaussian noise on the dynamic model with variances C_w.

For $k = 1$,

$$x_1 = s_1 + w_1,$$

and for $k > 1$

$$x_{k+1} = \Gamma_k^c x_k + w_{k+1}, \tag{3.13}$$

Under this hidden Markov model for class c the approximation of the log likelihood can be computed using equation (3.2).

3.5 Single and multiple point scatterers model for a bird

Let us now focus on the signal model for a bird, starting with a single point scatterer like we did in the UAV case and then we extend the model to a multiple point scatterer model, where both wings are modelled as lines of point scatterers. Once this model is derived we state the hidden Markov model and the likelihood function, which we need for the classifier.

To build the signal model for the bird we need to make assumptions that will simplify the modelling part. Assumptions A.1-A.6 still hold and additional assumptions are made.

A.10 The bird has a fixed orientation with respect to the radar and the orientation of the bird is parallel to the ground surface of the radar coordinate system, but the orientation in the $x-y$ plane is unknown, e.g. we do not know in which direction the bird's beak is pointing.

A.11 The angular wing position or elevation angle with respect to the bird is oscillating over time and is sinusoidal .

In literature this is the most common way of modelling the wing's movement [35].

A.12 The bird is flying, meaning flapping its wings.

For simplification we assume that the bird is only in flying mode, since gliding mode would be modelled differently.

Under assumptions A.1-A.6 and A.10-A.12 we have that point scatterer Q on the wing is mainly moving up and down with respect body of the bird. The wings are attached at center O located at (x_O, y_O, z_O) as given in (3.6). The location of point Q on the wing is expressed in the local coordinate frame with origin O and the same orientation as the radar coordinate frame, see Figure 3.2. The location of the rotating point in the local coordinate system is denoted by (x_Q, y_Q, z_Q) where

$$\begin{aligned} x_Q(t) &= l_Q \cos(\alpha_1) \cos(A_w \cos(\frac{2\pi t}{T_c} + \varphi_0)), \\ y_Q(t) &= l_Q \cos(\alpha_1) \sin(A_w \cos(\frac{2\pi t}{T_c} + \varphi_0)), \\ z_Q(t) &= l_Q \sin(A_w \cos(\frac{2\pi t}{T_c} + \varphi_0)), \end{aligned}$$

with initial elevation angle $A_w \cos(\varphi_0)$, range l_Q, i.e. the distance from the center O to the point scatterer Q, the duration of one wing stroke T_c, wing stroke amplitude A_w, which is the maximum angle the wing makes with respect to the surface and time t. The elevation angle of the point scatterer $A_w \cos(\frac{2\pi t}{T_c} + \varphi_0)$ is now oscillating over time between A_w and $-A_w$.

Now the distance between the point Q and radar as a function of time t is,

$$\begin{aligned} R_Q(t) &= \sqrt{(x_O + x_Q)^2 + (y_O + y_Q)^2 + (z_O + z_Q)^2} \\ &\approx R_0 + l_Q \left(\cos(\beta)\cos(\alpha - \alpha_1)\cos(\Lambda_k^c) + \sin(\beta)\sin(\Lambda_k^c)\right), \end{aligned} \tag{3.14}$$

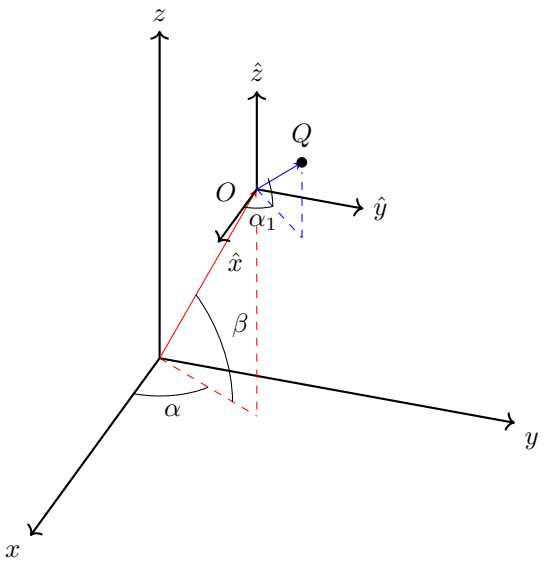


Figure 3.2: The fixed center O is placed at range R_0 and with azimuth angle α and elevation angle β in the radar coordinate system. There is a local coordinate frame $(\hat{x}, \hat{y}, \hat{z})$ with center O in which the point scatterer Q is expressed with respect to origin O.

where

$$\Lambda_k^c = A_w \cos(\frac{2\pi t}{T_c} + \varphi_0).$$

The justification for the approximation in (3.14) is similar to the approximation in 3.3 and is given in Appendix B.4.

Next we can make the signal model. Substituting (3.14) into (3.4) gives us the received signal for a bird as a function of the class dependent variables: duration of one wing stroke T_c and length of wing L_w and amplitude of wing stroke A_w and as a function of radar/sensor variables: initial phase Φ_0, amplitude $A(t)$, angular carrier frequency $\omega_d = 2\pi f_d$ and wavelength $\lambda = \frac{c}{f_d}$

$$s_{R_Q}(t) = \rho A(t) e^{-j[\omega_d t + \Phi_0]} e^{j\frac{4\pi}{\lambda}[R_0]} e^{\frac{j4\pi}{\lambda} l_Q (\cos(\beta)\cos(\alpha-\alpha_1)\cos(\Lambda_k^c)+\sin(\beta)\sin(\Lambda_k^c))}. \tag{3.15}$$

This is model for the received signal at time t. Similarly as we have done with the previous signal model we build a hidden Markov model using the model (3.15). This hidden Markov model is used to calculate the log likelihood.

This single point scatterer model for a bird is the base for the full wing model, where we assume that the wing of a bird can be modelled as a wire of point scatterers as in Assumption A.7. The wing starts at length L_1 from the center of body O and has length L_w, hence if we

integrate over the full wing i.e. from L_1 to $L_2 = L_1 + L_W$ we get,

$$\begin{aligned} s_{\text{wing}}(t) &= \int_{L_1}^{L_2} s_{R_Q}(l_Q) dl_Q, \\ &= \rho A(t) e^{-j[\omega_d t + \Phi_0]} e^{j\frac{4\pi}{\lambda}[R_0]} \cdot L_W e^{\frac{j4\pi}{\lambda}\frac{L_1+L_2}{2}(\cos(\beta)\cos(\alpha-\alpha_1)\cos(\Lambda_k^c)+\sin(\beta)\sin(\Lambda_k^c))} \\ &\quad \cdot \operatorname{sinc}\left(\frac{4\pi}{\lambda}\frac{L_W}{2}\left(\cos(\beta)\cos(\alpha-\alpha_1)\cos(\Lambda_k^c)+\sin(\beta)\sin(\Lambda_k^c)\right)\right). \end{aligned}$$

For the full bird we assume two wings having the same elevation angle all the time and second wing is located and the opposite side of the center O so the local azimuth angle is shifted π rad with respect to the other wing. Adding the two wing models together gives us the double wing model

$$\begin{aligned} s_{\text{bird}}(t) = L_W \rho A(t) e^{-j[\omega_d t + \Phi_0]} e^{j\frac{4\pi}{\lambda}[R_0]} \cdot \sum_{n=1}^{2} e^{\frac{j4\pi}{\lambda}\frac{L_1+L_2}{2}(\cos(\beta)\cos(\alpha-\alpha_1-n\pi)\cos(\Lambda_k^c)+\sin(\beta)\sin(\Lambda_k^c))} \\ \cdot \operatorname{sinc}\left(\frac{4\pi}{\lambda}\frac{L_W}{2}\left(\cos(\beta)\cos(\alpha-\alpha_1-n\pi)\right)\cos(\Lambda_k^c)+\sin(\beta)\sin(\Lambda_k^c)\right)\right). \end{aligned}$$

The double wing model is most extended model for the bird we will use in this thesis. Next the hidden Markov model is derived for the double wing model.

If $|s_k| > 0$ the hidden Markov model is of the form,

$$x_1 = s_1^c + w_1,$$

and for $k > 1$

$$x_{k+1} = \Gamma_k^c x_k + w_k, \tag{3.16}$$

where,

$$\Gamma_k^c = \frac{s_{\text{bird}}(t_{k+1})}{s_{\text{bird}}(t_k)}.$$

If we have $|s_k| = 0$, we will add a small positive number ϵ to the denominator to prevent dividing by zero.

Further we assume additive zero mean circular complex Gaussian noise on the dynamic model with variances C_w.

For this type of hidden Markov model as seen in the UAV model, we can compute the log likelihood using equation (3.2) which is used to build the classifier.

In the next chapter we investigate the performance of the classifier for the single and multiple point scatterers model and see what the limitations of these classifiers are.

Chapter 4

Numerical results

In this section we test the classifier based on the models developed in section 3.3, 3.4 and 3.5 and find the limitations. We test the performance of the classifier under different levels of noise and see how sensitive the classifier is to biases in the underlying dynamic models. The performance of estimation and learning of parameters is also investigated. For the numerical analysis we make the following assumption.

A.13 The range R_0, azimuth angle α, elevation angle β and fixed height z_0 are known.

The range R_0, azimuth angle α and elevation angle β are parameters that can accurately be determined by existing radar techniques e.g. parallel tracking, therefore these parameters are assumed to be known. The parameter z_0 is an offset in the z-axis which can be assumed known without loss of generality.

We start by analysing the classifier which is based on the single point scatterer models as derived in section 3.3 for a UAV and in section 3.5 for a bird. Finally we analyse the classifier which is based on the multiple point scatterers models, the rotor model, derived in section 3.4, and the double wing model, derived in section 3.5, for the UAV and bird respectively. Recall that the classifier is using the likelihood function of all classes that arise from the hidden Markov models.

4.1 Single point scatterer models

We generate synthetic data according to the hidden Markov models derived in sections 3.3 and 3.5. The corresponding classifier classifies the observations into one class, which is represented by the class dependent variables.

Each class is characterised by a set of class dependent parameters. For example we consider two subclasses of UAVs distinguished by different values of rotations speed ω_c and rotation radius l_P. Similarly we consider two subclasses of birds identified by the unique combination of radius l_Q and length of period for one stroke T_w. See Table 4.1 and Table 4.2. We consider four classes, two UAVs and two birds such that we can compare how the classification between birds and UAVs is performing and to what extent the classifier can distinguish between birds or UAVs themselves.

c	Name	T_w	l_Q
1	Bird 1	0.3 s	0.3 m
2	Bird 2	0.1 s	0.2 m

Table 4.1: Bird dependent parameters where c is the number of the class.

c	Name	ω_c	l_P
3	UAV 1	500 rad/s	0.2 m
4	UAV 2	300 rad/s	0.3 m

Table 4.2: UAV dependent parameters where c is the number of the class.

We assume that all parameters are known with parameter values as in Table 4.3 and 4.4 and we generate 100 observations for each class using the hidden Markov models. The performance of the classifier is given in a confusion matrix in Figure 4.1.

Parameter	Value	Unity
A	5	m
f_d	10^8	$(1/s)$
Φ_0	$(1/2)\pi$	rad
C_v	0.1	-
f_s	50000	1/s
T_{tot}	0.1	s

Table 4.3: Sensor parameter values

The interpretation of these results are as follows. The rows correspond to the predicted class (Output Class), and the columns show the true class (Target Class). The diagonal cells show for how many and what percentage of the observations are classified correctly. The off diagonal cells show where the classifier has made misclassification. The column on the far right of the plot

Parameter	Value	Unity
ρ	1	-
R_0	10	m
β	$(1/3)\pi$	rad
α	$(1/4)\pi$	rad
z_0	0.1	m
C_w	0.01	-

Table 4.4: Other parameters

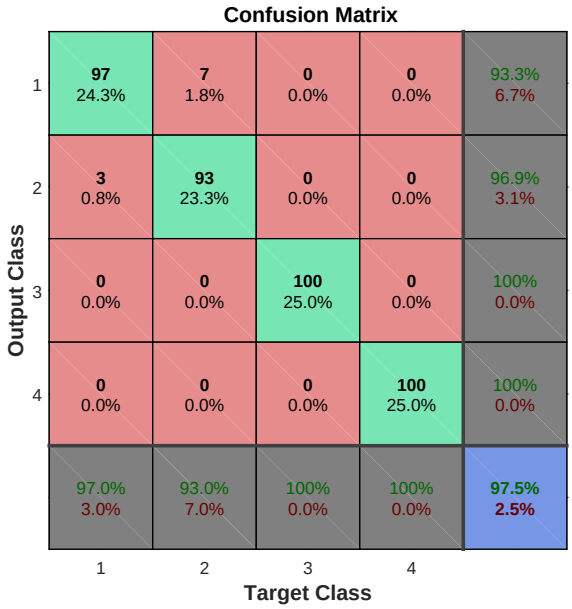


Figure 4.1: This confusion matrix is the result of the classification under the parameter values in the Tables 4.1, 4.2, 4.3 and 4.4.

shows the performance for each predicted class, while the row at the bottom of the plot shows the performance for each true class. The cell in the bottom right of the plot shows the overall performance. From all observations that are classified in class 1 (97+7=104) , $\frac{97}{104} \approx 93.3\%$ is classified correctly and from all observations from target class 1 $\frac{97}{100} = 97\%$ is classified correctly. The overall correct classification percentage is $\frac{97+93+100+100}{400} = 97.5\%$ (bottom right cell) , which is also the average percentages of correctly classified observations.

UAVs are classified completely correctly including the subclasses. On the other hand, though the birds are classified as birds, there are misclassification within the subclasses.

4.1.1 Sensitivity to measurement noise variance C_v

The results in Figure 4.1 were found for $C_v = 0.1$. In this section we look at the performance for different values of measurement noise variance C_v. In Figure 4.2 the overall performance, the percentage of correctly classified observations, is plotted over the different values of the measurement noise variance C_v. For each value of C_v we did ten iterations, to give a notion of the variance of performance for the same set of parameter values. In Figure 4.2 we see that even

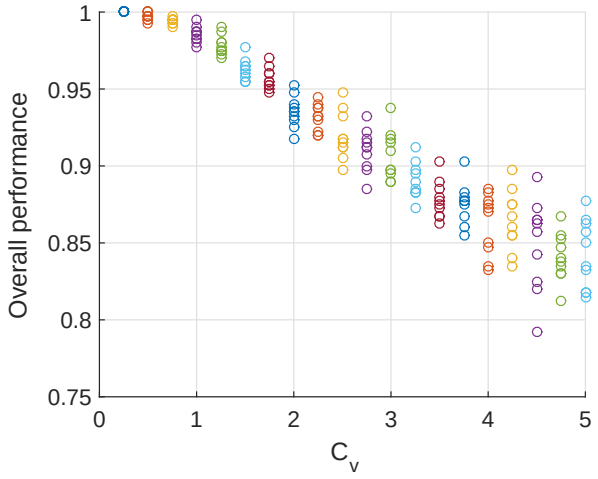


Figure 4.2: This figure shows the degradation of the performance if the measurement noise variance C_v increases. The observations originate from the 4 classes and the corresponding one point models as described above. For each value of C_v ten iterations are done to indicate the variance in the performance of the classifier for the same set of parameter values.

for large values of variance the overall performance is above 80%. In the corresponding confusion matrix (see Figure A.0.1 in Appendix A) we observe that in between UAVs misclassification occurs and between birds, but no bird is classified as an UAV or vice versa. In Figure 4.3 we see 4 observations $y_{1:K}$ of the four classes. For example all the red dots are sequential measurements of class Bird 2. The previous remark that UAVs are not misclassified as birds is not surprising if we look at Figure 4.3, since there is almost no overlap between the birds and UAV observations. Although the observations of the two different birds overlap, the classifier can classify them (in at least 80% of the observations) correctly using the underlying hidden Markov model.

4.1.2 Unknown initial phase φ_0

In the last section we assumed all parameters to be known, but in reality there are a few unknown parameters. One of unknown parameters is the initial phase of the single point scatterer with respect to the local coordinate frame and for the bird the initial phase is expressed as $A_w \cos(\varphi_0)$ where φ_0 is the unknown parameter. In this section we assume that all parameters are known except for the initial phase. This parameter can be estimated by the maximum likelihood estimation as discussed in section 2.4.3

$$\hat{\varphi}_0^c = \underset{\varphi_0 \in [0,2\pi]}{\arg\max}\, p(y_{1:K}|\varphi_0, c), \tag{4.1}$$

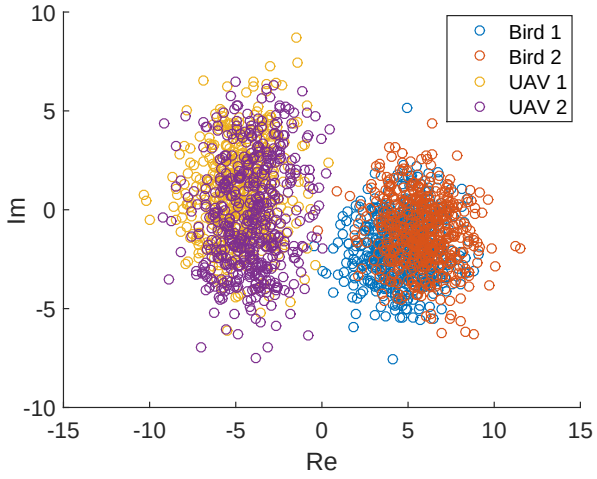


Figure 4.3: Four observations of the four classes are plotted. We see that the observations originating from the birds are almost separable from the UAV observations.

where we use the grid space $[0 : 0.01 : 2\pi]$ for estimation.

If we estimate the initial phase and classify using these estimates we observe an overall performance of 70% for $C_v = 0.1$. One of the confusion matrices is shown in Figure 4.4. Compare this result with the result in the previous situation where the initial phase was known, Figure 4.1. We see that introducing an extra uncertainty results in a significant decrease in overall performance (right bottom cell) of roughly 30% point (from 97.5% to 67.0%).

Investigation of the performance of the MLE shows that MLE is also not performing well for the initial phase. The likelihood has often a maximum at two different values. This is due to the symmetry of the cosine function in the model. This often leads to a large error between true and estimated value.

4.1.3 Learning process noise variance C_w

In the previous section a parameter was estimated for each observation $y_{1:K}$. In this section we assume the process noise variance to be unknown. This parameter is not observation dependent and can therefore be learned by the MLANS algorithm as described in section 2.4.1. This learning is done under two different assumptions. First we assume that the process noise variance is identical for all classes and secondly we assume that the process noise variance is class dependent. The underlying truth is that the process noise variance is class dependent. The prior probabilities are assumed to be known.

We generate unsupervised observations according to the above parameters, where C_w^c is different for each class.

The results discussed below are presented in Table 4.5.

Under the assumption that the process noise variance C_w is a feature of all small flying objects, meaning it is identical for all classes, we see that the estimated is close to the mean C_w^c

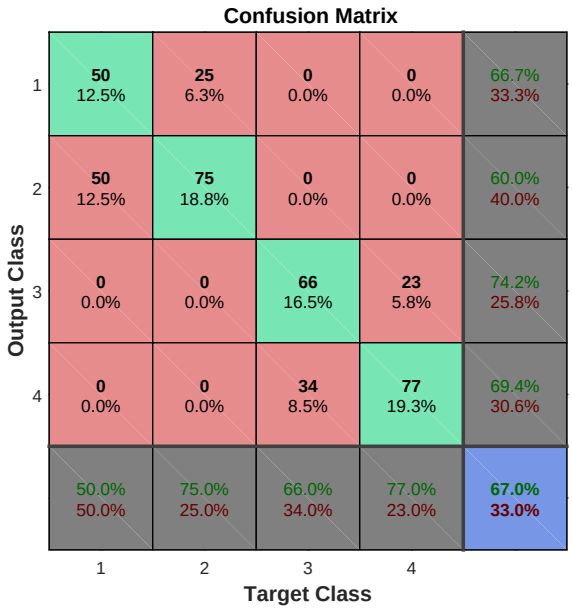


Figure 4.4: This confusion matrix is the results of classification after estimation of the unknown initial phase.

over all classes given in the third column of Table 4.5. The corresponding average overall performance of 95.9%, which is almost equal to the performance when all parameters are known and the process noise variance was identical for all classes. Again we observe no misclassifications between UAVs and birds, see Figure A.0.2 in the Appendix A.

Next we assume a different process noise variance for each class. The approximating likelihood gradient (4.2) with respect to C_w^c is

$$\frac{dll}{dC_w^c} = \sum_{n=1}^{N} P(c|y_{1:K}^n, C_w^c)\Big(\frac{-(K-1)}{(C_w^c+2C_v)} - \sum_{k=2}^{K}\left[\frac{|y_k^n|^2+|y_{k-1}^n|^2-2Re\left(y_k^n\overline{y_{k-1}^n}e^{-i\Phi_{k-1}^c}\right)}{(C_w^c+2C_v)^2}\right] + \frac{1}{C_w^c+C_v} + \frac{y_1^n-s_1^c}{C_w^c+C_v}\Big). \quad (4.2)$$

In Table 4.5 we see the estimates for the process noises variance (fifth column) that are estimated in iterative MLANS algorithm 1 where classification is improved significantly to 98.7% with respect to the case where was assumed that the process variance was identical for all classes (95.9%). This increase of performance with respect to the identical process noise variance is due to extra feature to distinguish between classes, namely process noise variance. We see that the estimates for the process noise variances are good, i.e. the estimates are deviating less than 4% from the true values.

		Identical C_w		Different C_w^c	
Parameter	True value	Estimate C_w	Performance	Estimate C_w^c	Performance
C_w^1	0.10	0.0655	95.9%	0.1013	98.7%
C_w^2	0.03			0.0291	
C_w^3	0.05			0.0485	
C_w^4	0.08			0.0791	

Table 4.5: The estimates and truth values for C_w using the MLANS algorithm.

4.2 Multiple point scatterers models

In the previous section the classifier was based on the single point scatterer models. In this section and the rest of this thesis the multiple point scatterers model are used: the rotor model (3.13) and the double wing model (3.16) are used for UAVs and birds respectively. The classifiers are based on the corresponding hidden Markov models. In this section we see how the classifiers are performing under different levels of noise and we investigate how sensitive the classifiers are to other uncertainties/biases on parameters in the hidden Markov model. First we define the class dependent parameters in the hidden Markov models and the rest of the parameter values.

In addition to the parameters that are present in the single point scatterer models, the rotor model includes the number of blades of the rotor N_b. Some of the parameters in the model are known such as the sensor parameters as given in Table 4.8. All class dependent parameters are assumed to take a finite number of values, since we consider a discrete class space and the space is assumed to be collectively exhaustive. Again we assume four classes, two UAVs and two birds. The UAV dependent parameters are given in Table 4.7, the bird dependent parameters in Table 4.6. So to be clear if we observe an UAV the length of the rotor blades is either 0.2 m or 0.3 m, there are no other UAVs observable.

c	Name	T_w	L_W
1	Bird 1	0.3 s	0.3 m
2	Bird 2	0.1 s	0.2 m

Table 4.6: Bird dependent parameters where c is the class number.

c	Name	ω_c	L	N_b
3	UAV 1	500 rad/s	0.2 m	4
4	UAV 2	300 rad/s	0.3 m	3

Table 4.7: UAV dependent parameters where c is the class number.

Parameter	Value	Unity
A	5	m
f_d	10^8	$(1/s)$
Φ_0	$(1/2)\pi$	rad
C_v	0.1	-
f_s	50000	1/s
T_{tot}	0.01	s

Table 4.8: Sensor parameter values

The rest of the parameters values are given in Tables 4.8 and 4.9 where the initial phase parameter φ_0 and the orientation of the bird α_1 are drawn randomly out of the uniform distribution on $[0, 2\pi]$ for each generated observation, but assumed known during the classification process. An observation for each class is plotted in Figure 4.5. Observe that three observations

Parameter	Value	Unity
ρ	1	-
R_0	10	m
β	$(1/3)\pi$	rad
α	$(1/4)\pi$	rad
z_0	0.1	m
C_w	0.01	-
A_w	$(1/3)\pi$	rad
φ_0	$\mathcal{U}(0, 2\pi)$	rad
α_1	$\mathcal{U}(0, 2\pi)$	rad

Table 4.9: Other parameters

have a lot of overlap and the observation from Bird 1 does not have much overlap with the other observations.

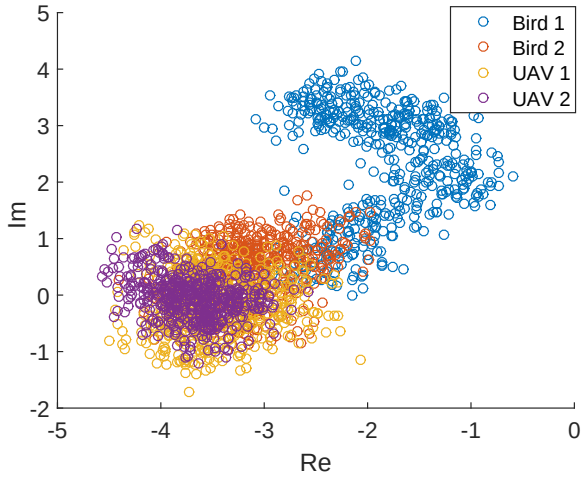


Figure 4.5: Four observations $y_{1:K}$ of the four different classes as described above. The observations lie in the complex plane and are generated using the hidden Markov models.

Given the parameters in Table 4.7, 4.8 and 4.9 we generate 100 observations with the hidden Markov model for each class. Next we classify the observations assuming that all parameter values are known. The overall performance of this classification is on average 84.3%. One confusion matrix is given in Figure 4.6. These results compared to the previous classification results based on the single point scatterer models lead to some remarks.

Under the multiple point scatterer models identifying the subclasses of UAVs becomes harder. Whereas in the single point scatterer case the classification of UAVs was perfect the classification of UAVs decreases to an average performance of 68.6%. In the confusion matrix in Figure 4.6 the performance of the classification of UAVs is $\frac{68+61}{200} = 64.5\%$. On the other hand we observe

that the performance of bird classification has increased slightly.

The decrease in performance of the UAV classification is justifiable. In the single point scatterer we received a signal that was only given information about this one point scatterer, in the rotor model the information of infinite point scatterers are reduced to one number. From this received signal it is harder to extract information about the scatterer from this signal.

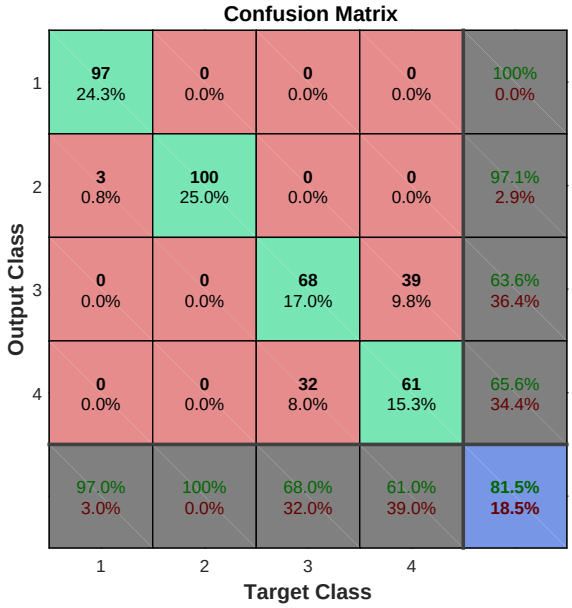


Figure 4.6: This confusion matrix is the result of the classification under the parameter values in the Table 4.6, 4.7, 4.8 and 4.9. All parameters are assumed to be known.

4.2.1 Sensitivity of measurement noise C_v

Next we investigate the sensitivity of the performance of the classifier with respect to the measurement noise variance C_v and observe the performance of the classifier under different variances.

The overall performance is plotted for different values of measurement noise in Figure 4.7. We observe from the confusion matrices (one confusion matrix is plotted in Figure A.0.4 in Appendix A for one confusion matrix) that for small values $C_v < 1$ the classifier is still able to distinguish between UAVs and birds, but as $C_v \geq 1$ the classifier is misclassifying birds as UAVs and vice versa.

Actually the value of the measurement noise variance should be compared to amplitude of the signal model to get a notion of relative noise. This can done by the signal to noise ratio (SNR). Since the signal amplitude remains constant we work with the absolute numbers for the noise variance. The amplitude of the signal is equal to $L\rho A = 1$, so the SNR (for example, $\frac{L\rho A}{C_v}$

is 10 for a variance of 0.1 or $10\log_{10}(\frac{L\rho A}{C_v}) = 10\ dB$).

Further analysis of the results in Figure 4.7 shows that the performance is more sensitive to change in noise variance compared to the classification performance under the single point scatterer models in Figure 4.2. The curve decreases faster as the noise variance increases. Mainly due to misclassifications of UAVs the performance curve is lower than the performance curve in Figure 4.2.

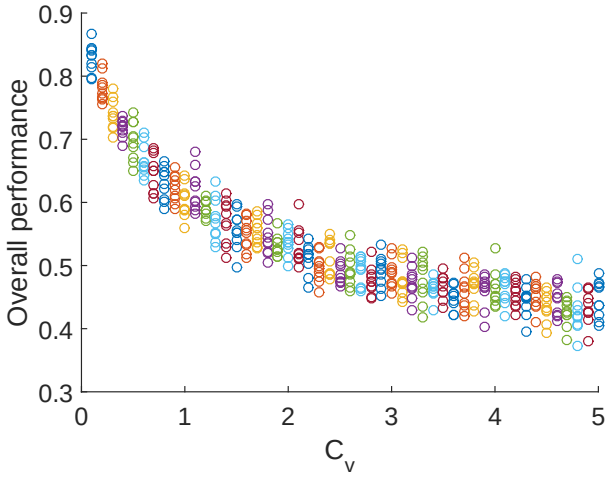


Figure 4.7: This figure shows the degradation of the performance if the measurement noise variance C_v increases. The observations originate from the four classes and the corresponding rotor and double wing models as described above. For each value of C_v ten iterations are done to indicate the variance in the performance of the classifier. The ten iterations are plotted in the same color.

4.2.2 Estimating initial phase φ_0

In the above section it was assumed that the initial phase of the wing or the rotor is known, but in reality this parameter is unknown since it depends on the moment of measuring. So in this section we assume that the initial phase is unknown and needs to be estimated.

If the estimation of the initial phase is done poorly, for example, when estimates are drawn randomly, we observe a significant drop in the performance with respect to the case where all parameters are known, see Figure A.0.5 in Appendix A for one of the confusion matrices. The decrease is roughly 25% point (from 84.3% to 60.1% performance).

In general the initial phase is different for each observation $y_{1:K}$. The observation $y_{1:K}$ is known prior to the estimation, so we estimate this parameter using the information $y_{1:K}$ and the maximum likelihood estimate as we have done before in section 4.1.2. The classification performance of this approach is given in the confusion matrix in Figure 4.8.

The average overall performance is decreasing to 75.3% (compare to the 84.3% when all parameters are known and the 60.1% when poor estimation is done) , but remarkable is the fact

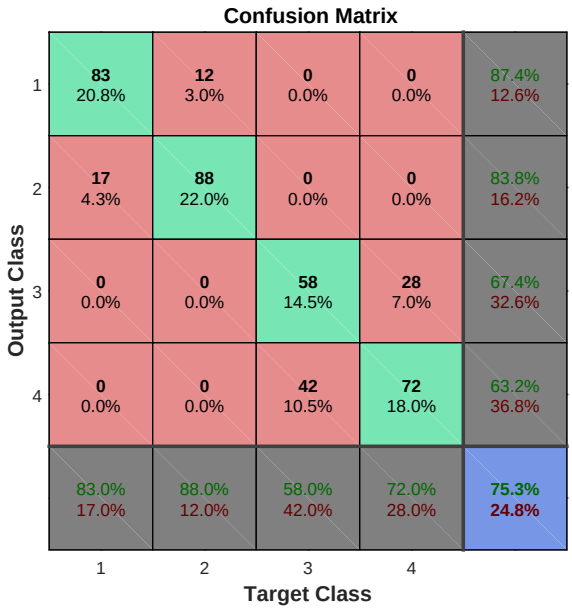


Figure 4.8: This confusion matrix is the result of the classification under the parameter values in the Tables 4.7, 4.8 and 4.9, where initial phase φ_0 is estimated using maximum likelihood estimation.

that the performance of UAV classification is not decreasing. Since the overall performance is decreasing we conclude that the bird classifications is accountable for this decrease in overall performance. Comparing the decrease in performance due to estimation of the initial phase between the single point scatterer model and extended models we see that the decrease is smaller for the latter model, namely only 10% point (from 84.3% to 75.3%) versus almost 30% point from 97.5% (Figure 4.1) to 67.0% (Figure 4.4) .

4.2.3 Learning process noise variance C_w and prior probabilities P_c

Previously we focussed on estimation of parameters that depend on one observation, so the values of these parameters are unknown and different for every observations. In this section we will apply the learning techniques as we did in section 4.1.3 for learning parameters which are class dependent. The parameter that is assumed unknown is again the process noise variance. The underlying truth is that the process noise variance is class dependent.

In the single point scatterer models classification we found an analytical solution for the optimal parameter value for C_w^c by equating the approximated likelihood gradient in (4.3) equal to zero. However, for the hidden Markov models developed from the extended models there are no tractable analytical solutions. Therefore the optimal solution can be found by a numerical

method, the gradient ascent method with gradient

$$\frac{dll}{dC_w^c} = \sum_{n=1}^{N} P(c|y_{1:K}^n, C_w^c)\Big(\sum_{k=2}^{K} \frac{-1}{(C_w^c + C_v + |\Gamma_{k-1}^c|^2 C_v)} - \sum_{k=2}^{K} \left[\frac{|y_k^n|^2 + |y_{k-1}^n|^2 - 2Re\left(y_k^n \overline{y_{k-1}^n \Gamma_{k-1}^c}\right)}{(C_w^c + C_v + |\Gamma_{k-1}^c|^2 C_v)^2} \right] + \frac{-1}{(C_w^c + C_v)} + \frac{-|y_1^n|^2 - |s_1^c|^2 + 2Re\left(y_1^n \overline{s_1^c}\right)}{(C_w^c + C_v)^2} \Big). \tag{4.3}$$

If we assume the process noise to be identical for all classes and start with initial guess for the process noise variance $\hat{C}_w = 0.1$ and for the prior probabilities $\hat{P}_c = [0.35\ 0.35\ 0.1\ 0.2]$ we find that the estimate is close to the average of the true class process noises variances as given in Table 4.10 (third column), the estimate for the prior probabilities are given in Table 4.11 (third column). The average performance decreases to 80.9% from the 84.3% in the case all parameters where known. One of the confusion matrix is shown in Figure A.0.6 in the Appendix A.

For the case where we assume that the process noises variances are different for each class where we start with as initial guess for the process noise variance $\hat{C}_w^c = [0.1\ 0.1\ 0.1\ 0.1]$ and for the prior probabilities $\hat{P}_c = [0.35\ 0.35\ 0.1\ 0.2]$, we find the process noises variances and the prior probabilities given in Table 4.10 (fifth column) and Table 4.11 (fourth column) respectively.

		Identical C_w		Different C_w^c	
Parameter	True value	Estimate C_w	Performance	Estimate C_w^c	Performance
C_w^1	0.05	0.0522	80.9%	0.1013	90.7%
C_w^2	0.03			0.0271	
C_w^3	0.1			0.1081	
C_w^4	0.2			0.2076	

Table 4.10: Estimates of the process class noise C_w^c under the assumption of identical variances (third column) and different variances (fifth column) using the MLANS technique given the parameters in Table 4.7,4.8 and 4.9

		Identical C_w	Different C_w^c
Parameter	True value	Estimate P_c	Estimate P_c
P_1	0.25	0.2540	0.2521
P_2	0.25	0.2460	0.2480
P_3	0.25	0.2346	0.2444
P_4	0.25	0.2654	0.2555

Table 4.11: Estimates of the prior probabilities P_c under the assumption of identical variances (third column) and different variances (fourth column) using the MLANS technique.

The average performance of classification over ten runs is 90.7%, compared to the case where we assume that all classes have identical process noise variance we see an increase with 10% point (from 80.9% to 90.7%). This high percentage of correct classifications is caused by the different value of C_w^c for each class. Note that in the previous simulations we assumed equal process variance for all classes. The different variances makes it easier to distinguish between models, since the likelihoods are more distinct.

For ten observations we have plotted the evolution over the number of iterations in Figure 4.9. The influence of the estimates for the prior probabilities and noise variances is observable. Initially (iteration 1), a few observations have a posterior probability smaller than 0.5, but as the parameters are learned we notice that initial misclassified observations are eventually classified correctly.

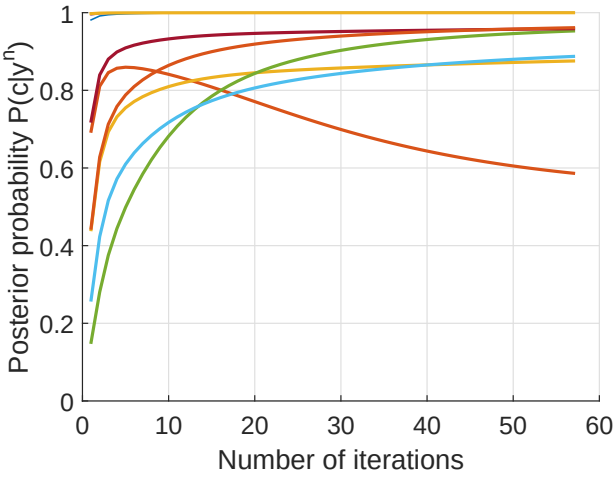


Figure 4.9: The evolution of the posterior probabilities for a few observations for one class c.

4.2.4 Estimating φ_0 and learning C_w and prior probabilities P_c

In last section we assumed the initial phase φ_0 to be known. In this section a more realistic approach is taken, where we assume the initial phase to be unknown. We combine the estimation and learning techniques to estimate values for parameters and see how this estimation and learning effects the classification performance.

The initial phase is estimated by maximum likelihood estimation over a fine grid, as we did before. The process variance is learned by the MLANS-technique are used in the classification. However, in general these approaches can not be taken separately, since the likelihood functions are dependent on the initial phase and the process variance. Therefore we will compute the MLE for the initial phase for each iteration in the MLANS-algorithm.

In the simulation results we observe that the estimates for the initial phase are not changing over the iterations in the MLANS algorithm as given in Figure 4.10. Therefore in the rest of the simulations both approaches were done separately, for the sake of computation time.

Again we generate observations according to the parameter values as given above, but we assume different process noises C_w^c for each class c. We want to learn these parameters with unlabelled observations. In Figure 4.11 we see the final classification performance of one simulation run, the learning process for the parameters C_w^c and P_c is shown in Figure 4.12 and Figure 4.13 respectively. The estimates of the prior probabilities are wrong for the UAVs. The prior probabilities do not converges to the true values, since there are still misclassifications as

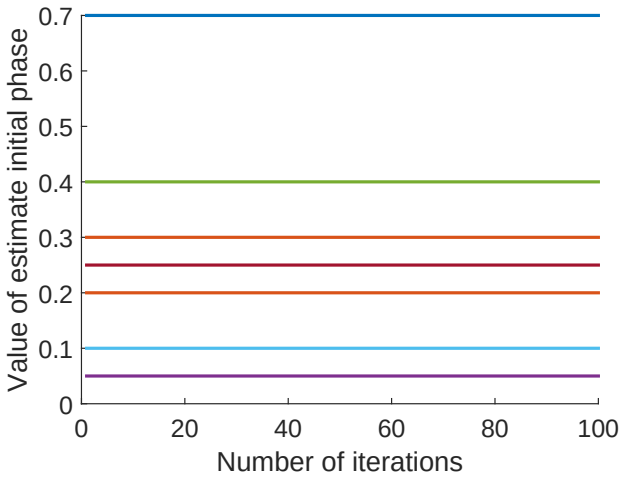


Figure 4.10: The maximum likelihood estimates are not changing over the iterations of the MLANS-algorithm. Each line represents the estimate for one observations.

one can see in Figure 4.13. The average performance is 72.3% over ten simulations runs and

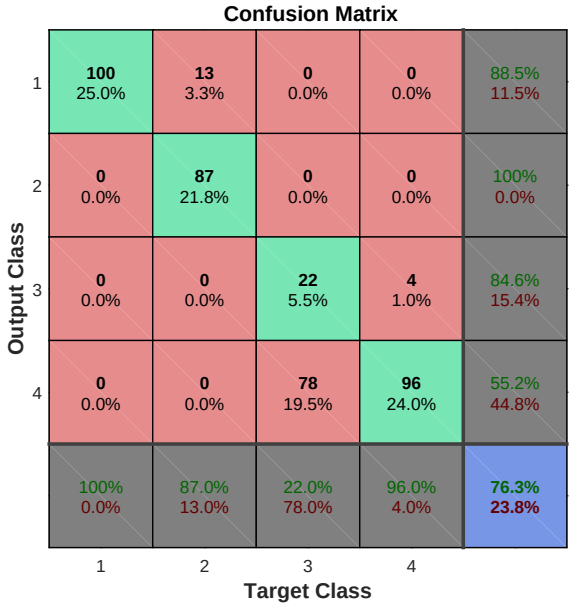


Figure 4.11: Classification results of 400 observations, where one unknown parameter is estimated and subsequently the process noise variance is learned.

is ranging from 45% to 90%. The performance is highly dependent on the estimates for the

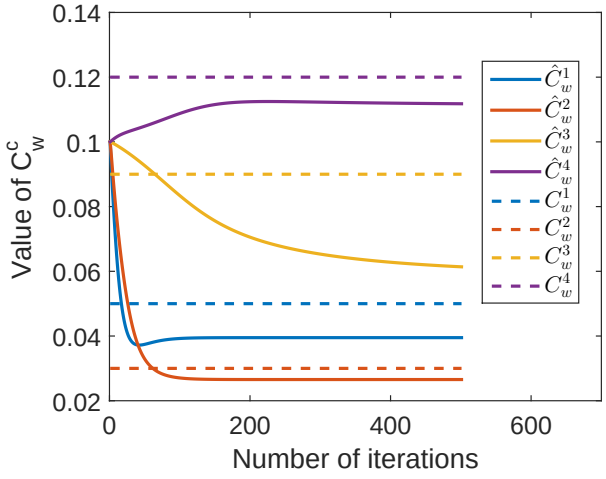


Figure 4.12: Convergence of the process noise variance. The true values are C_w^c plotted as dashed lines and the estimates $\hat{C}_w^c$ are the solid lines. The common initial value for the estimates is 0.1 and from there the estimates convergence are not close to the true values.

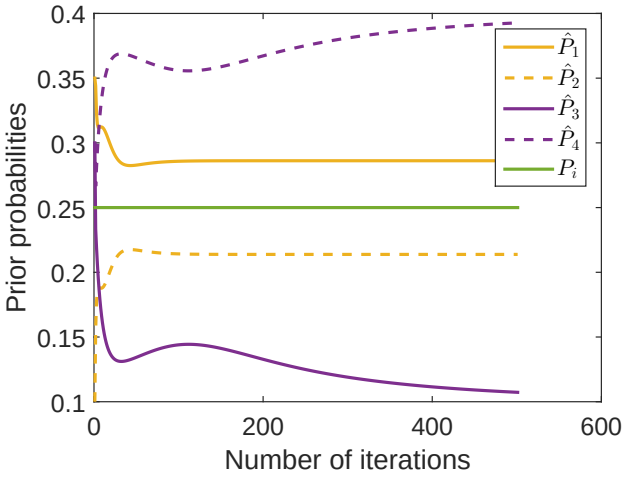


Figure 4.13: Convergence of the process noise variance. The true prior distribution is the discrete uniform distribution. The initial guess is $\hat{P}_c = [0.35\ 0.35\ 0.1\ 0.2]$ from which the estimates convergence towards values not close to the true value 0.25

process noise variance and prior probability estimates. The MLANS-technique does not always find the correct estimates for the process noise variances nor the prior probabilities, resulting in misclassification. Since the MLANS algorithm is an optimisation algorithm it can get stuck local maxima. The confusion matrices show that there are rarely misclassifications between the birds and UAVs, which is reassuring since the main goal is to differentiate between UAVs and birds. If the estimates are good the performance is high $(80 - 90\%)$ compared to the case where

all parameters are known.

4.2.5 Estimation of biases at other levels

The classifier is based on the models developed in section 3, but to test if the classifier is robust we assume more uncertainty in the hidden Markov models. This uncertainty is introduced by incorporating biases on parameter at different levels in the hidden Markov model. These biases are assumed to differ for every observation, but for every observation we assume them constant over time. We consider two biases, a bias on the amplitude parameter and a bias on the length of the blade/wing. First we will consider the bias on the amplitude parameter.

First we assume a bias δ_A^n on the amplitude of the signal $L\rho A$ of the n-th observation such that the modelled amplitude becomes $L\rho A + \delta_A$, all other parameters are assumed to be known. The bias δ_A^n is assumed to be normally distributed with zero mean and variance ϵ_A^c for the observation $y_{1:K}^n$ originating from class c. We compute the maximum likelihood estimates on a large grid. The variance of the bias ϵ_A^c equals 0.05 in the simulations. The performance of classification under these assumptions are shown in Figure 4.14 and we observe that the average overall performance over ten simulation runs is 74.0%, which is almost equal to the case where we assume the initial phase to be unknown (75.3%). We observe that a few UAVs are classified as birds and vice versa.

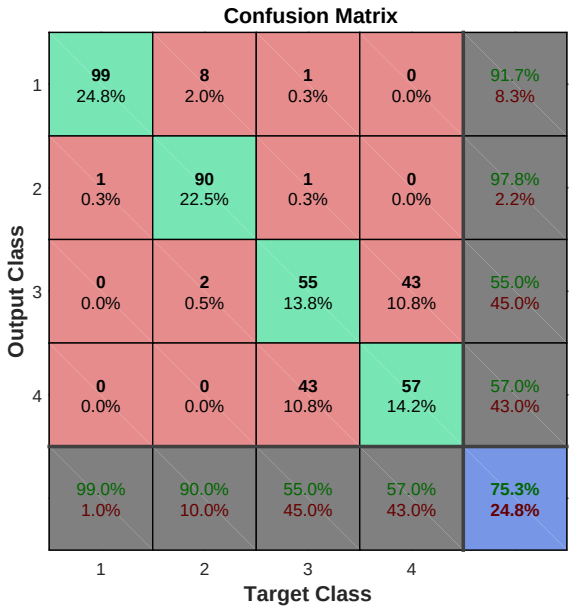


Figure 4.14: Classification performance under assumption that there is a bias on the amplitude parameter which is estimated. We now observe misclassification between UAVs and birds.

From Figure 4.15 we observe from these histograms that the estimation of the biases on the amplitude is performing poorly. If the biases were estimated correctly the histogram would

show a more normal distribution alike shape. However, the poor performance of estimation is not affecting the classification performance, which can be explained as follows. The classifier is based on the approximation likelihood, where the difference between two successive measurement is compared, it is not taking the amplitude into account for these successive measurements since the amplitude is assumed to be constant over time. Only the amplitude of the first measurement is compared to the amplitude of the underlying model, hence we can explain the poor performance of estimation. Conversely the performance of classification is not affected significantly due to the same reason, i.e. the classifier is classifying based on relative differences between measurements in the observation $y_{1:K}$.

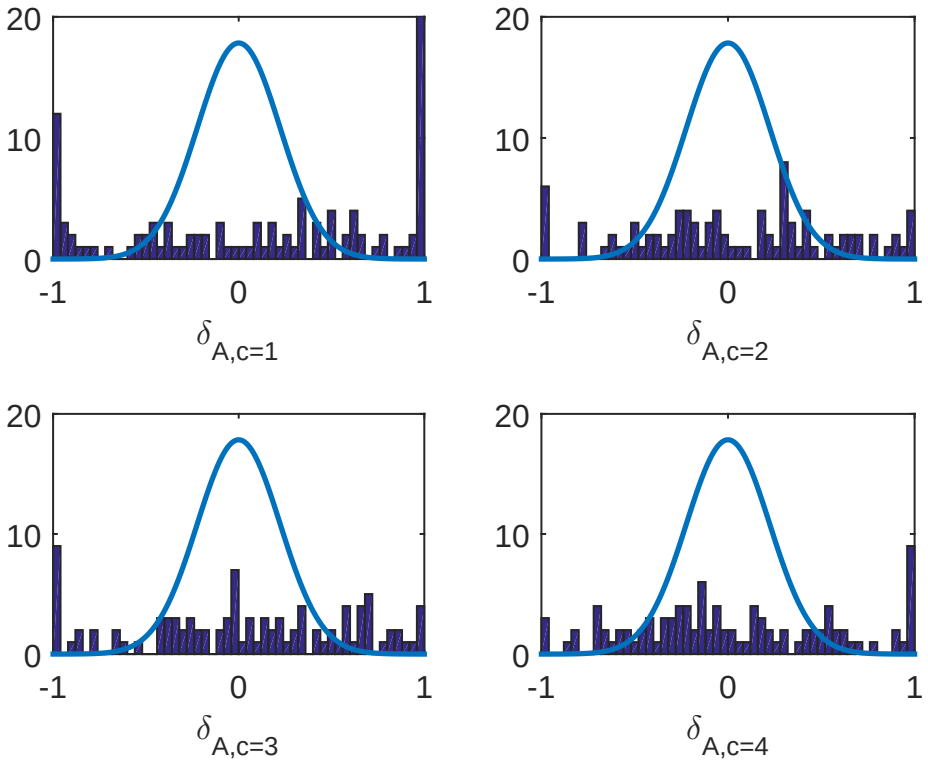


Figure 4.15: Histograms of estimates of the bias δ_A^n. One histogram for each class c with all observations that are classified into class c. The underlying true density function is also plotted.

The second bias is considered to be a bias on the length of the blade or wing. The bias, denoted by δ_L^n, is again at a different level of the model and we assume that the distribution of this bias is class dependent: $\delta_L^n \sim \mathcal{N}(0, \epsilon_L^c)$. First we will estimate this parameter value for each observation and see how this estimation influences the performance of the classifier. Subsequently we will see how the estimation is performing.

Again we assume all other parameters to be known with values given in Tables 4.6, 4.7, 4.8, 4.9 and $\epsilon_L^c = 0.05$. The biases δ_L^n are estimated and these estimates are then used in the classifier.

The performance of classification is averaged over ten simulation runs and equals an average performance of 70.5%. Again we observe that estimation of the biases is performing poorly. There is no significant difference between the error $|\hat{\delta}_L^n - \delta_L^n|$ of the misclassified observations and the error of the correctly classified observations. On the other hand if we classify under random estimates for the biases the performance is dropping to 40%, so although the estimates are not close the true values, it does improve the performance compared to a poor estimation, e.g. randomly generated estimates.

Compared to the case where all parameters are assumed to be known and the process variance was known and identical for all classes, the performance has decreases from 84.3% to 70.5%.

Next we assume biases on all three parameters we have investigated earlier, so a bias on the amplitude, length of the blade/wing and a bias on the initial phase. The biases on the amplitude and length will be normally distributed with variances $\epsilon_A^c = 0.05$ and $\epsilon_L^c = 0.05$ respectively and the initial phase will be uniformly distributed over $[0, 2\pi]$. Further we assume the parameter in Tables 4.6, 4.7, 4.8 and 4.9. Due to the larger parameter space the computation time has increased significantly and we observe that the performance decreases again to an average of 61.3%, where misclassifications are also in between UAVs and birds. But on average only 5% UAVs and birds are misclassified, meaning UAV classified as birds or vice versa. Decreasing the measurement noise variance from 0.1 to 0.01 increases the average overall performance to 70%. So the decrease in performance due to estimation of three unknowns is not equal to the sum of the decreases in performance separately.

4.2.6 Multiple classes

We introduce more classes as given in Tables 4.12 and 4.13 to investigate on which class dependent parameters the classifier can best distinguish between the classes and investigate its limitations in differentiation. Again under the assumption that all parameters are known as given in Table 4.6, 4.7, 4.8 and 4.9 and the process variance is identical for all classes.

c	Name	T_w	l_W
1	Bird 1	0.1 s	0.3 m
2	Bird 2	0.1 s	0.4 m
3	Bird 3	0.1 s	0.8 m
4	Bird 4	0.2 s	0.3 m
5	Bird 5	0.3 s	0.3 m

Table 4.12: Bird dependent parameters where c is the class number.

The parameters are chosen such that the influence of the parameters can be easily detected. The performance results in Figure 4.16 show that the classification based on a difference in rotation speed is hard, since a lot of observations from UAV 1, UAV 6 and UAV 7 are misclassified in between these classes. For UAVs we observe that classification based on a difference in blade length or number of blades is easier. Another remarkable observation is that UAV 8 with two blades is sometimes misclassified into a bird class and vice versa. This might be due to the fact that two blades are quite similar to two wings.

We notice that discriminating between birds with a small difference in wing length is hard, since Bird 1 and Bird 2 are misclassified mutually. A larger difference between Bird 1 and Bird

c	Name	ω_c	l_P	N_b
6	UAV 1	500 rad/s	0.2 m	4
7	UAV 2	500 rad/s	0.2 m	3
8	UAV 3	500 rad/s	0.2 m	2
9	UAV 4	500 rad/s	0.3 m	4
10	UAV 5	500 rad/s	0.5 m	4
11	UAV 6	450 rad/s	0.2 m	4
12	UAV 7	300 rad/s	0.2 m	4

Table 4.13: UAV dependent parameters where c is the class number.

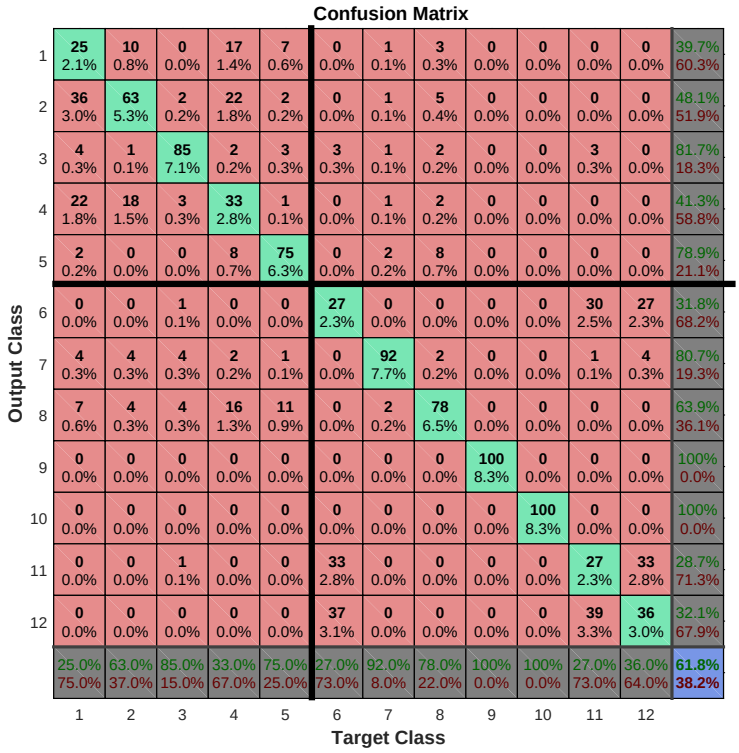


Figure 4.16: Classification performance with multiple classes as described in Table 4.12 and Table 4.13 under the assumption that all parameters are known.

3 can be used by the classifier to distinguish between both classes. Bird 1 and Bird 4 are often misclassified mutually, indicating that a small difference between stroke period of the wing T_w cannot be detected by the classifier. For the results we see that it is easier to distinguish between Bird 1 and Bird 5 due to a larger difference in period of wing stroke.

Note that these results are found under the parameters as given in Tables 4.12, 4.13, 4.8 4.9. For example, the small difference in stroke period is detectable when the total measurement time

T_{tot} is larger, e.g. one second.

The above observations are emphasized by the likelihood functions for these class parameters given in Figure 4.17. Where both rotation speed and one period stroke length show periodicity and the length and number of blades the likelihood decreases if the variable is further away from maximiser. The likelihood function over the length of the wing is not shown here, because the shape is similar to the likelihood function over the length of the blade.

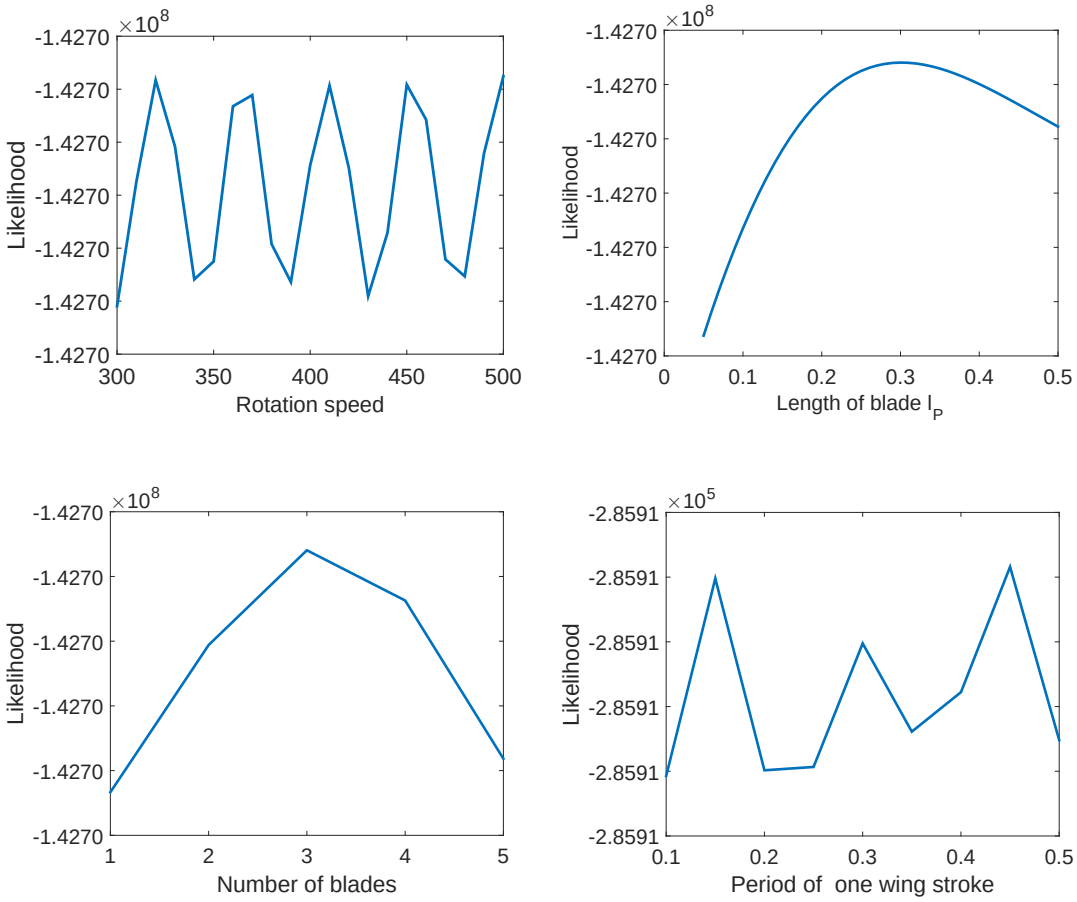


Figure 4.17: Likelihood function over different class dependent variables.

Mainly due to time constraints the sensitivity analysis was done for a few parameters. The parameters that were chosen were in our view the most interesting. The conclusions of the results presented in this chapter are summarised in the next chapter.

Chapter 5

Conclusions and Future work

In this thesis an automatic classification algorithm for birds and (mini-)UAVs was presented. The approach to build this classifier is a Bayesian approach where the underlying class models are assumed to be hidden Markov models. The MLANS algorithm is able to learn parameter values using unlabelled observations and can classify these unlabelled observations after learning.

The analysis of the performance and limitations of the classifier is performed with synthetic observations, generated from the models developed in Chapter 3, hence the conclusions are based on a synthetic set of data and it is not clear how the approach is relating to reality. Nevertheless there are a few conclusions that can be drawn from this analysis.

We have observed that classification under the assumption that underlying models are based on single point scatterers performs better than under the assumption that underlying models are multiple point scatterers models. In the latter case information of the multiple scatterers is reduced to one number at each moment in time.

The rest of the conclusions are given for the classifier based on the multiple point scatterers models, since most of the results were done under these multiple point scatterers models.

First we conclude that even under the assumption that we have to estimate biases on three different parameters in the model we are able to classify 61.3% of the observations correctly (in the case of the four given classes), where only 5% of the misclassifications are between UAVs and birds. Therefore the classification under the assumption with three uncertainties on the parameters classifies a bird or UAV correctly in 95% of the observations. Further we observed that even under a high level of measurement noise the classifier is able to distinguish between birds and UAVs, which is based on the different underlying dynamics both objects have.

Secondly we have concluded that although the classification was performing good under estimation of biases on parameters, the estimation itself was performing poor under high levels of noise. As expected estimation is performing well when the noise variance is small. This poor estimation might have been avoided if no approximation of the likelihood function was made. The approximation likelihood function does not take into account all information, what can have caused the poor estimation.

Thirdly we have worked with the unsupervised learning technique MLANS. Unsupervised

learning is beneficial since obtaining supervised observations is expensive. We can conclude that the MLANS algorithm is highly dependent on the initial estimates for the process noise variances and prior probabilities, just like other non convex optimisation problems. The algorithm often gets stuck in local maxima and this causes a decrease in performance. On the other hand if the initial estimates are good the MLANS approach does work and converges to the true underlying parameter values. The problem of avoiding local maxima is not within the scope of this project, but can help to improve the performance of the classifier.

Further, we have conclusions about the limitation of this classifier on classification of UAVs into subclasses of UAVs and birds into subclasses of birds. We can conclude from the results that the classification of UAVs based on the rotation speed is performing poor, especially when the observations is done in a small time window. On the contrary, we observed that it is more easy to discriminate between UAVs if the UAVs have a different number of blades or different length of blades. The limitations of distinguishing between class dependent variables is also strongly related to the likelihood function. When the analytical likelihood function is used, we can expect that the classifier is able to distinguish more easily between class dependent variables, meaning that the likelihood function is a more spiky function.

For birds we can conclude that it is hard to discriminate between birds based on both class dependent parameters: the period of one wing and the length of the wings. A small difference in wing length is not enough to distinguish between two birds. By observing a bird for a longer time, we see that discrimination based on the period of one wing stroke is going well.

Next we will discuss two direction of future work.

The first direction of future work should first focus on the verification of this approach with (expensive) real data. The classifiers is based on the theoretical return model, but we need to investigate how this model is relating to the reality, real data. The most extended model for a UAV in this thesis is the rotor model, but a UAV usually exist of multiple rotors and a body. Assumptions that are made to develop this model are not all realistic. For example the assumption that no shielding occurs is does not always hold. Hence the real data might be not similar to the synthetic data of one rotor as generated in this thesis. So it would be good to first check the model assumptions by comparing the real data with the synthetic data. Further we expect that an extension of the model will be a better fit with the real data, but an extension also means that information of more point scatterers is projected onto one number. Therefore we expect that the classification performance decreases if the underlying dynamic models become more complex.

The second direction of future work should focus on the improvement of performance of the parameter estimation. In this thesis the parameter estimation is performing poorly and this might be due to the approximation of the likelihood. First we recommend to compute maximum likelihood estimates using the exact likelihood, which can be calculated. Subsequently, if the estimation is performing better, we need to study the effect on the performance of classification. We have seen that if all parameters are known that the performance is 84.3%. The performance in the case of three unknowns was 61.3%, so there is a potential 20% performance to gain, which can be achieved by better parameter estimation.

Bibliography

[1] James O. Berger. *Statistical decision theory, foundations, concepts, and methods / James O. Berger*. Springer-Verlag New York, 1980.

[2] James Orvis Berger. *Statistical decision theory and bayesian analysis*. Springer series in statistics. Springer, New York, Berlin, Heidelberg, 1985. Autre tirage : 2010.

[3] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[5] Jerome S. Jerome Seymour Bruner. A study of thinking, 1956.

[6] V. C., F. Li, S. S. Ho, and H. Wechsler. Micro-doppler effect in radar: phenomenon, model, and simulation study. *IEEE Transactions on Aerospace and Electronic Systems*, 42(1):2–21, Jan 2006.

[7] Olivier Cappé, Eric Moulines, and Tobias Rydén. *Inference in Hidden Markov Models*, volume 48. 2005.

[8] Victor C. Chen. *Micro-Doppler effect in radar*. London: Artech House, 2011.

[9] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, January 1967.

[10] R. T. Cox. Probability, frequency and reasonable expectation. *American Journal of Physics*, 14(1), 1946.

[11] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989.

[12] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B*, 39(1):1–38, 1977. With discussion.

[13] Josh Elliot. Drone flight near vancouver airport attracts transport canada. `http://www.ctvnews.ca/canada/drone-flight-near-vancouver-airport-attracts-transport-canada-rcmp-attention-1.1788494`. Accessed: 2016-07-21.

[14] Robert J. Elliott, John B. Moore, and Lakhdar. Aggoun. *Hidden Markov models : estimation and control / Robert J. Elliott, Lakhdar Aggoun, John B. Moore*. Springer-Verlag New York, 1995.

[15] D. Gaglione, C. Clemente, F. Coutts, Gang Li, and J. J. Soraghan. Model-based sparse recovery method for automatic classification of helicopters. In *2015 IEEE Radar Conference (RadarCon)*, pages 1161–1165, May 2015.

[16] Zoubin Ghahramani. Hidden markov models. chapter An Introduction to Hidden Markov Models and Bayesian Networks, pages 9–42. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2002.

[17] T.P. Gill. *The Doppler effect: An introduction to the theory of the effect.* Logos Press, London, 1965.

[18] E T Jaynes. Bayesian methods: General background, 1986.

[19] R. Kleinman and R. Mack. Scattering by linearly vibrating objects. *IEEE Transactions on Antennas and Propagation*, 27(3):344–352, May 1979.

[20] K. Kosanovich, A. Gurumoorthy, E. Sinzinger, and M. Piovoso. Improving the extrapolation capability of neural networks. In *Intelligent Control, 1996., Proceedings of the 1996 IEEE International Symposium on*, pages 390–395, Sep 1996.

[21] Pat Langley. *Elements of Machine Learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.

[22] J. Martin and B. Mulgrew. Analysis of the theoretical radar return signal form aircraft propeller blades. In *Radar Conference, 1990., Record of the IEEE 1990 International*, pages 569–572, May 1990.

[23] G. J. McLachlan and D. Peel. *Finite mixture models.* Wiley Series in Probability and Statistics, New York, 2000.

[24] J. Misiurewicz, K. Kulpa, and Z. Czekala. Analysis of recorded helicopter echo. In *Radar 97 (Conf. Publ. No. 449)*, pages 449–453, Oct 1997.

[25] P Molchanov, K Egiazarian, J Astola, R I A Harmanny, and J J M De Wit. Classification of small UAVs and birds by micro-Doppler signatures. pages 172–175, 2013.

[26] Emanuel Parzen. On estimation of a probability density function and mode. *Ann. Math. Statist.*, 33(3):1065–1076, 09 1962.

[27] Leonid I. Perlovsky. *Neural Networks and Intellect Using Model-Based concepts.* 2001.

[28] Leonid I. Perlovsky and Ross W. Deming. Neural networks for improved tracking. *IEEE Transactions on Neural Networks*, 18(6):1854–1857, 2007.

[29] Elaine Rich. *Artificial Intelligence.* McGraw-Hill, Inc., New York, NY, USA, 1983.

[30] Ronald L. Rivest. Learning decision lists. *Mach. Learn.*, 2(3):229–246, November 1987.

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chapter Learning Internal Representations by Error Propagation, pages 318–362. MIT Press, Cambridge, MA, USA, 1986.

[32] H. Schneider. Application of an autoregressive reflection model for the signal analysis of radar echoes from rotating objects. In *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pages 1236–1239 vol.2, Apr 1988.

[33] EH Shortliffe. *Computer-based medical consultations: MYCIN*. American Elsevier, 1976.

[34] Ferdi Van der Heijden, Robert P. W. Duin, Dick De Ridder, and David M. J. Tax. *Classification, Parameter Estimation and State Estimation: An Engineering Approach Using MATLAB*. 2004.

[35] TORKEL WEIS-FOGH. Energetics of hovering flight in hummingbirds and in drosophila. *Journal of Experimental Biology*, 56(1):79–104, 1972.

[36] Richard Whittle. Drone skies: The unmanned aircraft revolution is coming. `http://www.popularmechanics.com/military/a9407/drone-skies-the-unmanned-aircraft-revolution-is-coming-15894155`. Accessed: 2016-07-21.

[37] C. F. Jeff Wu. On the convergence properties of the em algorithm. *Ann. Statist.*, 11(1):95–103, 03 1983.

[38] M.S. Zediker, R.R. Rice, and J.H. Hollister. Method for extending range and sensitivity of a fiber optic micro-doppler ladar system and apparatus therefor, December 8 1998. US Patent 5,847,817.

Appendix A

Confusion matrices

In this Appendix some of the exemplary confusion matrices are plotted to justify the claims that are made in Chapter 4. The captions of the figures below describe under what conditions the results were obtained.

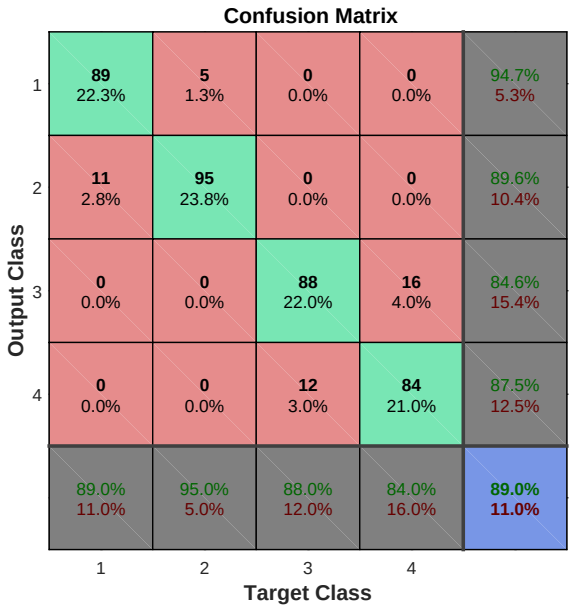


Figure A.0.1: This confusion matrix is the results of classification using underlying single point scatterer models and under the assumption that all parameter values are known with for $C_v = 4$

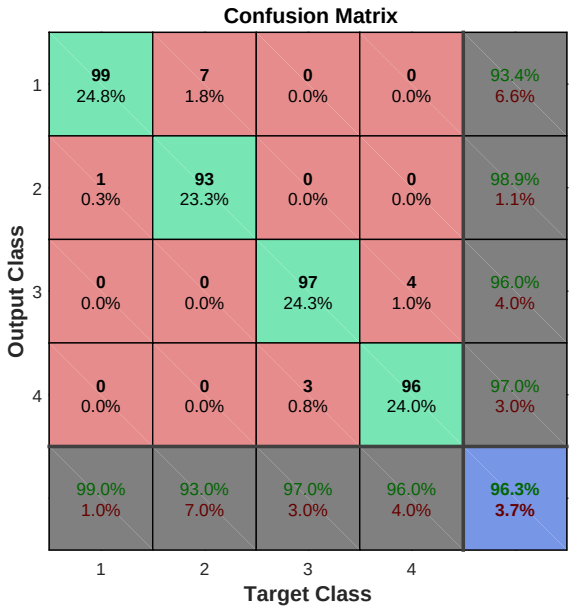


Figure A.0.2: This confusion matrix is the results of classification under learning C_w under the assumption that the process noise variance is identical for each class. These results are for the case where the underlying models are based on the single point scatterers.

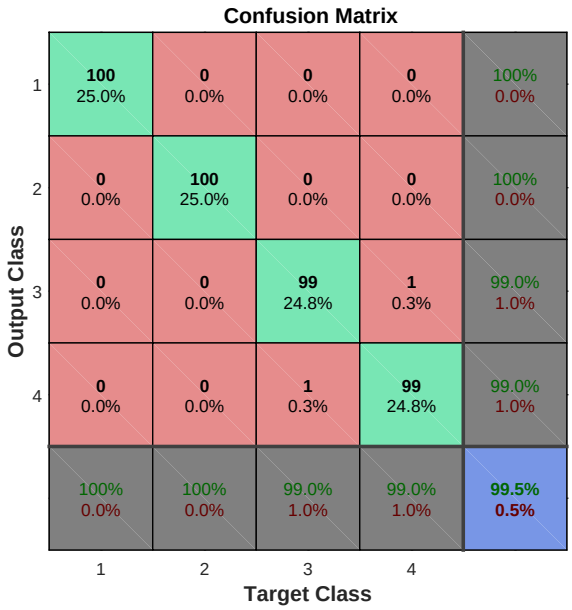


Figure A.0.3: This confusion matrix is the results of classification learning under the assumption that the process noise variance is different for each class with underlying single point scatterer models.

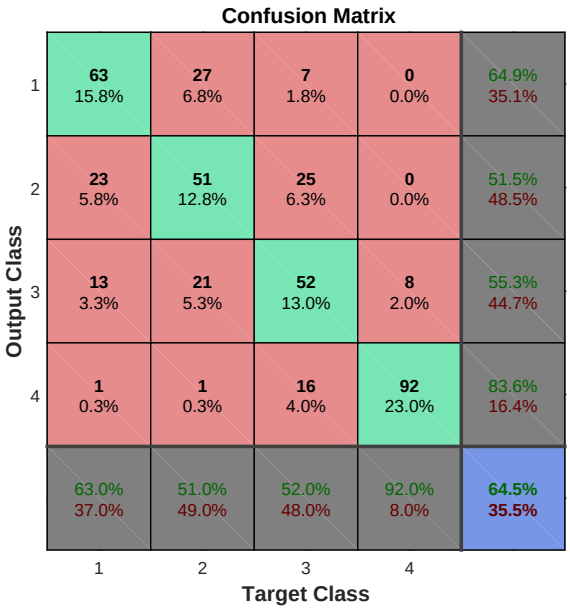


Figure A.0.4: This confusion matrix shows the results of classification using underlying multiple point scatterers model, where all parameters are known with $C_v = 2$.

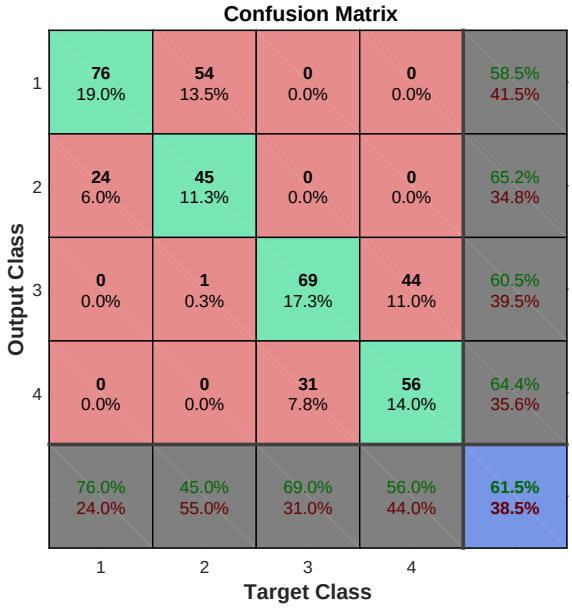


Figure A.0.5: This confusion matrix shows the results of classification using underlying multiple point scatterers model, where the initial phase is estimated by a randomly drawn estimate in $[0, 2\pi]$.

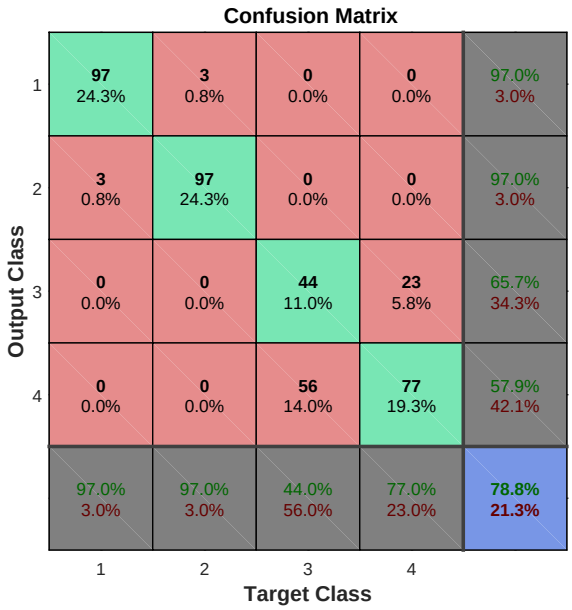


Figure A.0.6: This confusion matrix is the results of classification learning C_w and P_c under the assumption that the process noise variance is identical for all classes. The underlying models are the multiple point scatterer models.

Appendix B

Derivations

B.1 Derivation of the semi log-likelihood gradient

The semi-log-likelihood gradient is

$$ll'(\chi|\boldsymbol{y}) = \sum_{i=1}^{N} \log\left(\sum_{c\in C} p(y^i|c,\chi)P(c)\right) + \mu\left(\sum_{c\in C} P(c) - 1\right).$$

Differentiation of the semi-log-likelihood with respect to the model parameters $\chi_{\tilde{c}}$ for class $\tilde{c}$ gives

$$\begin{aligned}\frac{\partial ll'(\chi|\boldsymbol{y})}{\partial \chi_{\tilde{c}}} &= \sum_{i=1}^{N} \frac{\partial}{\partial \chi_{\tilde{c}}} \log\left(\sum_{c\in C} p(y^i|c,\chi)P(c)\right) \\ &= \sum_{i=1}^{N} \frac{\frac{\partial}{\partial \chi_{\tilde{c}}} \sum_{c\in C} p(y^i|c,\chi)P(c)}{\sum_{c\in C} p(y^i|c,\chi)P(c)} \\ &= \sum_{i=1}^{N} \frac{\frac{\partial}{\partial \chi_{\tilde{c}}} p(y^i|\tilde{c},\chi)P(\tilde{c})}{\sum_{c\in C} p(y^i|c,\chi)P(c)}.\end{aligned}$$

Using identity $\partial x = x \cdot \partial \log(x)$ we have

$$\begin{aligned}\frac{\partial ll'(\chi|\boldsymbol{y})}{\partial \chi_{\tilde{c}}} &= \sum_{i=1}^{N} \frac{p(y^i|\tilde{c},\chi)P(\tilde{c})}{\sum_{c\in C} p(y^i|c,\chi)P(c)} \frac{\partial}{\partial \chi_{\tilde{c}}} \log\left(p(y^i|\tilde{c},\chi)P(\tilde{c})\right) \\ &= \sum_{i=1}^{N} P(\tilde{c}|y^i,\chi) \frac{\partial}{\partial \chi_{\tilde{c}}} \log\left(p(y^i|\tilde{c},\chi)P(\tilde{c})\right).\end{aligned}$$

For the prior probability $P(\tilde{c})$ we can use the above result to find

$$\begin{aligned}\frac{\partial ll'(\chi|\boldsymbol{y})}{\partial P(\tilde{c})} &= \sum_{i=1}^{N} P(\tilde{c}|y^i,\chi) \frac{\partial}{\partial P(\tilde{c})} \log\left(p(y^i|\tilde{c},\chi)P(\tilde{c})\right) + \mu \\ &= \sum_{i=1}^{N} \frac{P(\tilde{c}|y^i,\chi)}{P(\tilde{c})} + \mu,\end{aligned}$$

B.2 Likelihood function

In this section we compute the likelihood of the observations given hidden Markov model for class c,

$$x_{k+1} = \Gamma_k^c x_k + w_k, y_{k+1} \qquad\qquad = x_{k+1} + v_{k+1},$$

where w and v are circular zero mean complex Gaussian noise with variance C_w and C_v respectively.

$$\begin{aligned} p(y_{1:K}|c) &= \prod_{k=1}^{K} p(y_k|y_{1:k-1}, c) \\ &= \prod_{k=1}^{K} \int p(y_k|x_k, c) p(x_k|y_{1:k-1}, c) dx_k \\ &= \prod_{k=1}^{K} \int \mathcal{CN}(y_k|x_k, C_v, c) \mathcal{CN}(x_k|\hat{x}_{k|k-1,c}, P_{k|k-1}, c) dx_k \qquad \text{(B.1)} \\ &= \prod_{k=1}^{K} \mathcal{CN}(y_k|\hat{x}_{k|k-1}, P_{k|k-1} + C_v, c), \qquad \text{(B.2)} \end{aligned}$$

We have,

$$p(x_k|y_{1:k-1}, c) = \mathcal{CN}(x_k|\hat{x}_{k|k-1}, P_{k|k-1}, c), \qquad \text{(B.3)}$$

since all noises are circular complex Gaussians. Therefore the real part and imaginary part are independent distributed and both parts can be modelled separately as linear Gaussian models. The Kalman scheme (B.4)computes the mean and variances for the corresponding Gaussian distributions. The real and imaginary part have identical variances if the noise is circular complex Gaussian and there we can add both parts together in one circular complex density function as given in (B.3)

Going from (B.1) to (B.2), we use again the fact that the real part and the imaginary part are both independently normal distributed and since v_k is independent from all other random components we can write (B.2).

B.2.1 Kalman filter

Given the linear Gaussian model

$$\begin{aligned} x_k &= F_k x_{k-1} + B_k u_k + w_k, \\ y_k &= H_k x_k + v_k, \end{aligned}$$

where Gaussian noises w_k, v_k have a zero mean normal distribution with variances Q_k and R_k respectively, the Kalman updating scheme is given by,

$$\begin{aligned}
\hat{x}_{k|k-1} &= F_k \hat{x}_{k-1|k-1} + B_k u_k, \\
P_{k|k-1} &= F_k P_{k-1|k-1} F_k^T + Q_k, \\
\tilde{y}_k &= y_k - H_k \hat{x}_{k|k-1}, \\
S_k &= H_k P_{k|k-1} H_k^T + R_k, \\
K_k &= P_{k|k-1} H_k^T S_k^{-1}, \\
\hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k, \\
P_{k|k} &= (I - K_k H_k) P_{k|k-1}.
\end{aligned} \tag{B.4}$$

B.3 Circular zero mean complex Gaussian noise

Circular complex Gaussian noise is complex noise for which the real and imaginary part are independently normally distributed with variance σ_w^2, mean μ and the probability density function is denoted by,

$$\begin{aligned}
\mathcal{CN}(w|\mu, C_w) &= \frac{1}{\pi C_w} e^{-\overline{(w-\mu)} C_w^{-1} (w-\mu)} \\
&= \frac{1}{\pi C_w} e^{-\frac{|w-\mu|^2}{C_w}},
\end{aligned}$$

where $w, \mu \in \mathbb{C}$, $C_w = 2\sigma_w^2 \in \mathbb{R}$, $|w|$ the modulus of w and $\overline{w}$ is the complex conjugate of w.

B.4 Approximation of distance model

To justify the approximation (3.8) we show that

$$\begin{aligned}
R_P^2(t) &\approx R_{app}^2(t), \\
&:= \left[R_0 + l_P \cos(\beta) \cos(\alpha - \omega_c t - \varphi_0) + z_0 \sin(\beta)\right]^2.
\end{aligned}$$

which implies that the squared error $(R_P(t) - R_A(t))^2$ is small. The absolute difference is,

$$\begin{aligned}
|R_P^2(t) - R_{app}^2(t)| &= |l_P^2 (1 - \cos^2(\beta) \cos^2(\alpha - \omega_c t - \varphi_0)) \\
&\quad + 2 z_0 \sin(\beta) l_P \cos(\beta) \cos(\alpha - \omega_c t - \varphi_0) - z_0^2 \sin^2(\beta)|, \\
&\leq (l_P + z_0)^2.
\end{aligned}$$

Since z_0 is a parameter that can be set to zero without loss of generality and $\frac{l_P}{R_0}$ is small we have that the relative error is also small.