MINIMUM HELLINGER DISTANCE CLASSIFICATION OF UNDERWATER ACOUSTIC SIGNALS

A Thesis in
Electrical Engineering
by
Brett Bissinger

© 2009 Brett Bissinger

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

December 2009
The thesis of Brett Bissinger was reviewed and approved* by the following:

Nirmal K. Bose  
HRB-Systems Professor of Electrical Engineering  
Thesis Co-Advisor

R. Lee Culver  
Associate Professor of Acoustics  
Thesis Co-Advisor

Richard Tutwiler  
Associate Professor of Electrical Engineering

W. Kenneth Jenkins  
Professor of Electrical Engineering  
Head of the Department of Electrical Engineering

*Signatures are on file in the Graduate School.
Abstract

Passive source classification in the underwater environment is a challenging problem in part because propagation through the space- and time-varying medium introduces variability and uncertainty in the signal. Acoustic propagation codes can predict received fields accurately but they are sensitive to input environmental parameters which cannot be known exactly. This uncertainty in environmental knowledge used in signal predictions results in imperfect statistical class models. Classifiers that rely on simulations of the environment must therefore be robust to imperfect environmental models. Maximum likelihood methods provide ideal performance when the class models are correct but their performance quickly deteriorates when class models are imperfect. Minimum distance methods generally can offer robustness to mismatches at the expense of performance, with that tradeoff governed by the distance metric used. Hellinger distance, when used as a distance metric, offers robustness to outliers while retaining the performance of a maximum likelihood method, properties that make it well-suited for classification of passive underwater acoustic signals. In the present work the robustness of the Minimum Hellinger Distance Classifier (MHDC) is quantified and its performance is compared to a Log-Likelihood Ratio Classifier (LLRC) with three different data sets: synthetic Gaussian data, synthetic acoustic data from propagation simulations and real acoustic data. In cases of acoustic data, class models are derived from Monte Carlo acoustic propagation simulations. In each case Receiver Operating Characteristic (ROC) curves show that the MHDC exhibits performance equivalent or superior to that of the LLRC, responding in a robust manner to imperfect class models.
# Table of Contents

List of Figures vii  
List of Tables ix  
List of Symbols x  
Acknowledgments xi

## Chapter 1  
### Introduction  
1.1 Motivation .......................................................... 1  
1.2 Thesis Overview and Organization ............................... 2

## Chapter 2  
### Minimum Hellinger Distance Classification  
2.1 Minimum Distance Classification ................................... 3  
2.1.1 General Structure .................................................. 3  
2.1.2 Distances In Statistics ........................................... 5  
2.1.3 Examples Of Divergences And Their Properties .................. 6  
2.1.4 The Generalized Gaussian To Examine Differences ............. 7  
2.1.5 Why Hellinger Distance ........................................... 10  
2.2 The Minimum Hellinger Distance Classifier ....................... 10  
2.3 Comparison of MHDC to Bayesian Methods ......................... 11  
2.3.1 Brief Introduction To Bayesian Methods ....................... 11  
2.3.2 MHDC vs. LLRC .................................................. 13  
2.3.3 ROC Curves Based On Gaussians .................................. 15  
2.4 Performance On Gaussian Data ..................................... 22  
2.4.1 Ideal Performance ................................................. 22  
2.4.2 Contamination Models ............................................. 23  
2.4.3 Contamination Results ............................................. 26  
2.5 MHDC With Density Estimation ...................................... 28
Chapter 3
Application to Synthetic Acoustic Data

3.1 Introduction ............................................. 30
3.2 Acoustic Simulations ................................... 30
  3.2.1 The Propagation Code - RAM ...................... 30
  3.2.2 The Acoustic Environment ......................... 31
  3.2.3 Pre-processing of RAM’s Output ................... 33
3.3 Received Data ........................................... 33
  3.3.1 Sequential Samples ................................. 35
  3.3.2 Interleaved Sequential Samples .................... 36
  3.3.3 Uniform Random Samples ......................... 37
  3.3.4 Results of Sampling Study ....................... 37
3.4 Models .................................................. 40
  3.4.1 Construction of Models ......................... 40
3.5 Results .................................................. 40
  3.5.1 Known Depth, Known Environment ............... 42
  3.5.2 Unknown Depth, Known Environment .......... 42
  3.5.3 Known Depth, Unknown Environment .......... 43
  3.5.4 Unknown Depth, Unknown Environment ....... 43
3.6 Conclusions ............................................. 45

Chapter 4
Application to Real Acoustic Data

4.1 Introduction ............................................. 46
4.2 The SwellEx-96 Experiment ............................ 46
  4.2.1 Setup for SwellEx-96 .............................. 46
  4.2.2 Acoustic Model ................................. 47
4.3 Processing .............................................. 49
  4.3.1 Pre-processing Of Received Signals ........... 49
  4.3.2 Detection Statistics And ROC Curves .......... 49
4.4 Results and Conclusions ................................ 50

Chapter 5
Conclusions

5.1 Summary ............................................... 56
5.2 Future Research ........................................ 57
  5.2.1 Metrics ........................................ 57
  5.2.2 Models ........................................ 57

Appendix A
Density Estimation

A.1 Kernel Density Estimates .............................. 58

Appendix B
Matlab Code

B.1 Forming models from TL fields ....................... 62
B.2 Running the classifiers .............................. 65
B.3 Calculating $p_{fa}$ and $p_d$ ......................... 68
List of Figures

2.1 Flow diagram for a general binary minimum distance classifier .......................... 4
2.2 Plots of the 3 parameter generalized Gaussian ................................................. 8
2.3 Distances between Gaussian and generalized Gaussian PDFs ............................... 9
2.4 Zoomed portion of Fig. 2.3 .................................................................................. 9
2.5 Flow diagram of the MHDC .................................................................................. 10
2.6 Ideal score in MHDC ............................................................................................. 11
2.7 Actual score in MHDC ........................................................................................... 12
2.8 Flow diagram for the LLRC ................................................................................... 13
2.9 A typical pair of probability density functions ..................................................... 14
2.10 Illustration of threshold, \( p_d \) and \( p_{fa} \) ................................................................. 15
2.11 A typical ROC curve ............................................................................................. 16
2.12 Underlying densities for Gaussian test data ......................................................... 17
2.13 Typical realizations of a Gaussian random variable ............................................. 17
2.14 Log-likelihood ratio function calculated from \( p_1 \) and \( p_2 \). ................................. 18
2.15 Probability density functions estimated from \( x_1 \) and \( x_2 \) for use as MHDC models. 18
2.16 Detection statistic results of a Monte Carlo simulation of the LLRC ..................... 19
2.17 Score results of a Monte Carlo simulation of the MHDC ....................................... 19
2.18 Probability density functions for detection statistics for the LLRC ....................... 20
2.19 Estimated probability density functions for detection statistics for the MHDC ... 20
2.20 A representative ROC curve for the LLRC’s performance on Gaussian data ...... 21
2.21 A representative ROC curve for the MHDC’s performance on Gaussian data ... 21
2.22 ROC curves showing the effect of realization size on classifier performance .... 22
2.23 ROC curves showing the effect of number of Monte Carlo runs on classifier performance ................................................................. 23
2.24 ROC curves showing the effect of changing variance on the performance of classifiers ................................................................. 24
2.25 Cartoon demonstrating inliers and outliers ......................................................... 25
2.26 Performance of the MHDC and LLRC with inliers .............................................. 27
2.27 Performance of the MHDC and LLRC with outliers ............................................ 28
2.28 Flow diagram for the full MHDC with probability density function estimation ... 29

3.1 GENLMIS acoustic environment ............................................................................ 32
3.2 Variations on the GENLMIS acoustic environment .............................................. 33
3.3 Transmission Loss (TL) as output of RAM ............................................................ 34
3.4 Transmission Loss in area of interest ...................................................................... 34
3.5 TL with ranged-based spreading corrected ............................................................ 35
3.6 Diagram of deep and shallow received data ......................................................... 36
3.7 Results of data sampling experiments ................................................................... 38
3.8 PDF of TL with N=10 subsections ........................................... 38
3.9 PDF of TL with N=100 subsections ........................................ 39
3.10 PDF of TL with N=1000 subsections ..................................... 39
3.11 TL data included in the known depth situation ...................... 41
3.12 TL data included in the uncertain depth situation ................. 41
3.13 Models produced for different levels of knowledge of the source and environment 42
3.14 Results for known depth and known environment .................. 43
3.15 Results for unknown depth and known environment ................ 44
3.16 Results for known depth and unknown environment ............... 44
3.17 Results for unknown depth and unknown environment .......... 45
4.1 SwellEx-96 Experiment Diagram ........................................... 47
4.2 SwellEx-96 simplified environmental model ........................... 48
4.3 Sample RAM TL output for SwellEx-96 simplified environment model .................. 48
4.4 MHDC detection statistics with interleaved sampling (N=500) .... 50
4.5 LLRC detection statistics with interleaved sampling (N=500) ..... 51
4.6 MHDC detection statistics with sequential sampling (N=500) .... 51
4.7 LLRC detection statistics with sequential sampling (N=500) ..... 52
4.8 ROC curves for 8 data sets using LLRC with sequential sampling . 52
4.9 ROC curves for 8 data sets using LLRC with interleaved sampling . 53
4.10 LLRC OC curves with combined data sets ............................. 54
4.11 MHDC OC curves with combined data sets ........................... 54
4.12 MHDC and LLRC performance on SwellEx data set ............... 55

A.1 A weighted sum of kernels forms a probability density function in a kernel density estimate .................................................. 60
A.2 Histograms contrasted with kernel density estimates .................. 60
List of Tables

2.1 Properties of a metric .................................................. 4
2.2 Examples of $\phi$-divergences ....................................... 5
2.3 Comparison of MHDC and LLRC ................................. 14
2.4 Statistics of contaminated LLRC detection statistic probability density functions 26
List of Symbols

$MHDC$  Minimum Hellinger Distance Classifier
$LLRC$  Log-Likelihood Ratio Classifier
$PDF$  Probability density function
$TL$  Transmission Loss
$RAM$  Range Acoustic Model for acoustic propagation

$f(x), g(x)$  Generic probability density functions
$\hat{f}(x)$  Estimate of the probability density function $f(x)$
$H_N$  Hypothesis number $N$
$p_N(x)$  Probability of choosing $H_N$
$x_N$  Realization of $p_N$
$\Delta$  Detection statistic (or score) that results from the MHDC
$\lambda(x)$  Likelihood ratio
$\eta$  Threshold in a binary classification test
$E\{\cdot\}$  Expectation operator
$d(\cdot, \cdot)$  A distance measure
$\phi(\lambda)$  Functional defining a $\phi$-divergence
$p_d$  Probability of detection
$p_{fa}$  Probability of false alarm
$ROC$  Receiver Operating Characteristic
$\epsilon$  Contamination factor in a contamination model
$c(z)$  Sound speed as a function of depth
Acknowledgments

I would like to thank my family for their support throughout my life in everything I do. They have always encouraged me to explore anything and everything and I wouldn’t have the wonderful wealth of experiences I have without that.

I would also like to thank my advisors Dr. N.K. Bose and Dr. Lee Culver for their support in my graduate studies. Their encouragement and guidance has been fundamental to my success. The third member of my committee, Dr. Rick Tutwiler, also deserves thanks for providing careful review and feedback. I would like to thank the Office of Naval Research for their support.

Finally I would like to thank my fellow graduate students for their friendly advice and camaraderie.
Chapter 1

Introduction

1.1 Motivation

Source classification and localization in the underwater environment is a challenging problem in part because propagation through the space- and time-varying medium introduces uncertainty in the signal. A variety of models describing the uncertainty and variability in ocean acoustic propagation, some of them quite complex, have been proposed including some recent work like [1] but the problem is still open. These models do not tend to lend themselves to integration with classification algorithms due to their implementation complexity and computational requirements, both significant drawbacks to practical use. A simpler model is therefore needed. The statistics of the amplitude of a received signal provides a model that is relatively simple to compute and store and is conducive to integration with a classifier.

Acoustic propagation codes can be used to predict acoustic fields very well but they are sensitive to input environmental parameters, which in practice cannot be known exactly. Uncertainty in the environment yields imperfect models, requiring robustness to mismatch between a modeled environment and the true environment. Bayesian processors operating on the uncertain ocean as in [2] can overcome small mismatches but are still too sensitive for practical use.

A minimum divergence classifier compares the divergences between the probability density functions of the received signal and predicted probability density functions corresponding to different classes, choosing the class whose probability density function has the smallest divergence from the received signal. This method of classification is known as statistical distance-based classification, a subset of minimum distance methods. Statistical distance-based methods have been used in estimation problems previously as in [3]. [4] showed that these methods can be adapted to classification applications as well.

The particular choice of the measure of statistical difference determines several important properties of the classifier. A family of statistical distance measures, independently proposed in both [5] and [6], including the well-known Kullback-Leibler divergence, Hellinger distance and
Battacharyya distance, provide a set of candidate statistical distances with attractive properties. As discussed in [7] and [8], Hellinger Distance, when used in a Minimum Distance Classifier, provides suitable optimality and robustness properties when models are imperfect.

In this thesis a binary depth classification is performed on underwater acoustic data where a received signal is classified as originating from either a deep or shallow source using the Minimum Hellinger Distance Classifier (MHDC) with class probability density functions derived from acoustic simulations.

1.2 Thesis Overview and Organization

In Chapter 2 the Minimum Hellinger Distance Classifier (MHDC) is developed. Minimum distance classification is first defined. A discussion of possible distances follows, with particular focus on divergence metrics. The Hellinger Distance is then defined and its properties are explained. The MHDC is then compared to a Bayesian processor in a theoretical sense. The practical steps needed to implement both the MHDC and a Bayesian processor are then presented with an explanation of the processing required to create the classifier’s class models and the required preprocessing of a received signal. The characteristics of the MHDC and its performance compared to a Bayesian processor on generated Gaussian data are then presented.

In Chapter 3 the performance of the classifier is evaluated on synthetic acoustic data. The setup of the acoustic simulations is presented along with some examples of simulated acoustic fields. The performance of the MHDC is then evaluated by varying the degree of mismatch between the environment used to generate the models and the environment used to generate a received signal to simulate an approximate knowledge of the environment in a controlled setting. The results of the MHDC are compared to a Bayesian processor, demonstrating the attractive properties of the MHDC.

In Chapter 4 the classifier is evaluated when applied to received data from the SwellEx-96 experiment and an appropriate model of its acoustic environment. SwellEx-96 and the environmental model are explained. The pre-processing required for the data set is then described. The MHDC’s performance is then compared to that of a Bayesian processor in a parallel of the analysis in Chapter 3.

In Chapter 5 conclusions are made and prospects for future work are explored.

Appendix A contains a brief overview of density estimation methods used in this work.
Chapter 2

Minimum Hellinger Distance Classification

2.1 Minimum Distance Classification

2.1.1 General Structure

A minimum distance classifier assigns an input to the class that minimizes a classification metric. Minimum Distance Classification may also be called nearest-neighbor or nearest prototype classification because the class chosen is that which is the closest to the input in a feature space. A feature space has axes called features that represent some aspect of the input. If one were classifying people, features could include height and weight. Each person would occupy a point in the feature space called . The two classes to be chosen from are "boy" called $H_1$ for hypothesis one and "girl" called $H_2$ for hypothesis two. To quantify this classification, a class model could be defined for "boy," called $M_1$ for model one, which would be a point in the feature space in the tall and heavy area of the space. A class model for girl, called $M_2$, could likewise be defined in the short and light section. With these two models, a person would be classified as a boy if closer to the boy model and as a girl if closer to the girl model. The term closer must be defined for this comparison to make sense. In this example the features are all numerical, so the Euclidean distance between the two points in space could be used as the distance metric, denoted $d(x, y)$. There are a variety of different metrics that can be used to define a distance in a space and this point will be revisited later in this chapter. With the distance defined in the feature space, a way to test whether a person is a boy or girl would be to calculate the difference between their distance to boy, defined as $d(M_1, D)$ and their distance to girl, defined as $d(M_2, D)$, and call that a score, which is defined as

$$\Delta = d(M_2, D) - d(M_1, D)$$ (2.1)
If it is thought that the boundary between girls and boys is the point directly between the two, $D$ would be assigned to class $M_1$, meaning hypothesis $H_1$ is true, whenever $\Delta > 0$ and $D$ would be assigned to class $M_1$, meaning hypothesis $H_2$ is true, whenever $\Delta < 0$. The boundary itself would not matter and could be assigned to either class. If, however, the boundary is not at the midpoint, a threshold $\eta$ would represent the boundary and replace zero in the previous inequalities. Fig. 2.1 shows a graphical representation of this classifier.

![Flow diagram for a general binary minimum distance classifier](image)

**Fig. 2.1.** Flow diagram for a general binary minimum distance classifier

There are several tasks involved in implementing such a classifier. These include mapping the data to a feature space, defining a distance on that feature space, and choosing the threshold. What follows is a discussion of these tasks for the construction of a minimum distance classifier for underwater acoustic signals.

To facilitate minimum distance applications, the distance should be at its minimum when the distance is between two like objects. Secondly, to conform to the traditional concept of distance, the distance should be non-negative. Therefore the distance from an object to itself should be 0 and the distance between two objects should only be 0 if the two objects are the same. A third desirable property is symmetry, where the distance from $x$ to $y$ is the same as the distance from $y$ to $x$. Last, a distance should add according to the triangle inequality, i.e. the distance from $x$ to $y$ plus the distance from $y$ to $z$ should be greater than or equal to the distance from $x$ to $z$. This allows distances between objects to be compared easily and reinforces the traditional concept of distance. These four properties define a "metric" in mathematical terms which can be summarized as follows:

<table>
<thead>
<tr>
<th>Property</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-negativity</td>
<td>$d(x, y) \geq 0$</td>
</tr>
<tr>
<td>Identity</td>
<td>$d(x, y) = 0$ iff $x = y$</td>
</tr>
<tr>
<td>Symmetry</td>
<td>$d(x, y) = d(y, x)$</td>
</tr>
<tr>
<td>Triangle Inequality</td>
<td>$d(x, z) \leq d(x, y) + d(y, z)$</td>
</tr>
</tbody>
</table>
2.1.2 Distances In Statistics

In the underwater acoustic environment, a received signal may be described by the probability density function of its amplitude. This leads to a feature space of probability density functions. It is then necessary to define distance in this space. The rest of this section discusses some desirable properties of the distance to be chosen, followed by some candidates for the distance, concluding with a choice of distance and a discussion of some of the practical aspects of the implementation of a minimum distance classifier based on this feature space and distance.

Armed with the general properties of the distance laid out in Table 2.1, the particulars of the feature space we have chosen (probability densities) need to be addressed. One way to examine the distance between two probability density functions is a function of the likelihood ratio of the two distributions. The likelihood ratio of two probability density functions, also known as the divergence between them, is defined as

$$\lambda(x) = \frac{g(x)}{f(x)}$$  \hspace{1cm} (2.2)

A classic definition of the distance between probability densities that uses their divergence is the $\phi$-divergence defined similarly by [5] and [6] as

$$d_{\phi}(f, g) = E_{F(x)}[\phi(\lambda(x))] = \int_X \phi(\lambda(x)) dF(x) = \int_X \phi\left(\frac{g(x)}{f(x)}\right) f(x) dx$$  \hspace{1cm} (2.3)

where $F(x)$ is a cumulative distribution function, $f(x) = \frac{dF(x)}{dx}$ and $g(x)$ are probability density functions, and $\phi(\lambda)$ is a convex function for $\lambda > 0$ with $\phi(1) = 0$. With this definition, the choice of $\phi(\lambda)$ determines most of the properties of the distance. Some well-known distances that have been derived in other manners come easily from this definition with a careful choice of $\phi(\lambda)$. Table 2.2 contains some examples.

<table>
<thead>
<tr>
<th>Name</th>
<th>$\phi(\lambda)$</th>
<th>$d_{\phi}(f, g) = E_{F(x)}[\phi(\lambda)]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kullback-Leibler</td>
<td>$\lambda \log(\lambda)$</td>
<td>$d_{KLS}(f, g) = d_{KL}(f, g) + d_{KL}(g, f)$</td>
</tr>
<tr>
<td>Symmetric Kullback-Leibler</td>
<td>$(\lambda - 1) \log(\lambda)$</td>
<td>$d_{BC}(f, g) = \int \sqrt{f(x)g(x)} dx$</td>
</tr>
<tr>
<td>Bhattacharyya</td>
<td>$\sqrt{\lambda}$</td>
<td>$d_{BC}(f, g) = \int \sqrt{f(x)g(x)} dx$</td>
</tr>
<tr>
<td>Squared Hellinger</td>
<td>$\frac{1}{2}(\sqrt{\lambda} - 1)^2$</td>
<td>$d_{2H}(f, g) = \frac{1}{2} \int (\sqrt{f(x)} - \sqrt{g(x)})^2 dx$</td>
</tr>
</tbody>
</table>

Each choice of $\phi(\lambda)$ possesses different properties and defines a distance on the space of probability distributions. Many papers ([9, 10, 11, 7, 8] as a small subset of examples) have been written on different divergences, both in the Ali-Silvey[5]/Csiszar [6] context and otherwise. This thesis will focus on $\phi$-divergences, but it is good to be aware of others.
2.1.3 Examples Of Divergences And Their Properties

The Kullback-Leibler (KL) divergence, also known as directed divergence, relative entropy and information divergence among other names, is defined as

\[ d_{KL}(f, g) = \int_X g(x) \log\left(\frac{g(x)}{f(x)}\right) dx \]  

(2.4)

which corresponds to a \(\phi\)-divergence of

\[ \phi(\lambda) = \lambda \log(\lambda) \]  

(2.5)

This KL divergence is non-negative and is zero only when its arguments are equal. It is not symmetric and does not obey the triangle inequality. It is often seen in communications and message coding applications. Its roots in information theory are explored in [12].

A logical alternative or extension to KL divergence is the Symmetric Kullback-Leibler divergence. It is the sum of the KL divergence calculated in both directions. Some definitions add a factor of a half to normalize the value, but it is not strictly required. It is defined as

\[ d_{KLS}(f, g) = d_{KL}(f, g) + d_{KL}(g, f) \]  

(2.6)

which corresponds to a \(\phi\)-divergence of

\[ \phi(\lambda) = (\lambda - 1) \log(\lambda) \]  

(2.7)

and it exhibits the first two properties of a metric like the KL divergence but is also symmetric. It does not, however, obey the triangle inequality. Its use is similar to the KL divergence.

A third \(\phi\)-divergence is the Bhattacharyya distance. It is defined as

\[ d_{BC}(f, g) = \int_X \sqrt{f(x)g(x)} dx \]  

(2.8)

which corresponds to a \(\phi\)-divergence of

\[ \phi(\lambda) = \sqrt{\lambda} \]  

(2.9)

It satisfies the first three properties of a metric, but does not obey the triangle inequality. A nice feature of the Bhattacharyya distance is its limited range. Indeed, its range is limited to [0, 1], making it quite attractive for a distance comparison. Kailath in [11] uses a derivative of the Bhattacharyya distance to simplify calculations in some detection theory applications.

A fourth \(\phi\)-divergence is the squared Hellinger distance. The squared Hellinger distance is
defined as
\[ d^2_H (f, g) = \frac{1}{2} \int_X \left( \sqrt{f(x)} - \sqrt{g(x)} \right)^2 \, dx \]  \tag{2.10}

which corresponds to a \(\phi\)-divergence of
\[ \phi(\lambda) = \frac{1}{2} \left( \sqrt{\lambda} - 1 \right)^2 \]  \tag{2.11}

The Hellinger distance itself is then defined as
\[ d_H (f, g) = \frac{1}{\sqrt{2}} \sqrt{\int_X \left( \sqrt{f(x)} - \sqrt{g(x)} \right)^2 \, dx} \]  \tag{2.12}

This quantity exhibits all four properties of a metric with a range of \([0, 1]\). This makes it an ideal candidate for estimation and classification problems. Minimum Hellinger Distance (MHD) methods have been well explored in estimation problems, starting with [3] and more recently including [13] and [14] among others. These methods generally estimate an unknown parametric density from data, using Hellinger distance as a performance metric. In addition to applications in estimation, the robustness properties of MHD methods has been well-explored in [7] and [8] among others.

### 2.1.4 The Generalized Gaussian To Examine Differences

Some of the properties of the different metrics introduced in the previous section can be demonstrated with a graphical analysis using a three parameter Generalized Gaussian distribution.

#### 2.1.4.1 The Generalized Gaussian

The generalized Gaussian distribution as defined in [12] has three parameters: mean \((\mu)\), standard deviation \((\sigma)\) and shape \((\alpha)\). It is defined in terms of the gamma function \(\Gamma(\cdot)\) as
\[ p(x; \mu, \sigma, \alpha) = \frac{1}{2\sigma A} e^{-\frac{|x-\mu|^{\alpha}}{A}}, A = \sqrt{\frac{\sigma^2 \Gamma(1/\alpha)}{\Gamma(3/\alpha)}} \]  \tag{2.13}

The mean and standard deviation have the same meaning as for the standard Gaussian distribution. The shape parameter \(\alpha \geq 0\) controls the overall shape of the distribution and shows the utility of the distribution. Fig. 2.2 shows the generalized Gaussian plotted with \(\mu = 0, \sigma = 1\) and the shape parameter \(\alpha\) varied from 0 to 99.

Several values of the shape parameter converge to other common distributions. With \(\alpha = 0\), the generalized Gaussian becomes a Dirac delta function and for \(\alpha = 1\) the generalized Gaussian becomes a Laplacian distribution. For \(\alpha = 2\) the generalized Gaussian becomes a Gaussian distribution and finally as \(\alpha \to \infty\) the generalized Gaussian approaches a Uniform distribution.

In general a small shape parameter corresponds to a peaked distribution and a large shape
Fig. 2.2. Plots of the 3 parameter generalized Gaussian with zero mean, unity variance and varying shape parameter $\alpha$ parameter corresponds to a broad distribution.

2.1.4.2 Distance Measures Applied To The Generalized Gaussian

The ability of a distance measure to distinguish between two similar distributions can be demonstrated with the generalized Gaussian by varying the shape parameter of a generalized Gaussian continuously and finding its distance from a generalized Gaussian with $\alpha = 2$ (corresponding to a Gaussian). Fig. 2.3 shows a comparison between three distance measures with $\alpha$ varied from 0 to 8. All means are zero and all standard deviations are one. Each distance measure is normalized so that its maximum value is 1. The results for the symmetric Kullback-Leibler divergence, the Bhattacharyya distance and the Hellinger distance are presented. The Kullback-Leibler divergence is omitted, as its characteristics are quite similar to the symmetric version. This normalization allows the shape to be analyzed without the distraction of the relative scale of the distances.

The Hellinger distance shows a very different characteristic from the other two distances, especially around $\alpha = 2$. Fig. 2.4 shows the area around $\alpha = 2$ in more detail.

It can be seen that the Hellinger distance provides the sharpest minimum of the three distances. This is useful in classification as the Hellinger distance will show the most difference between two probability densities that only vary a small amount, as is expected for underwater acoustic probability densities.
Fig. 2.3. Three different types of distances between Gaussian and generalized Gaussian probability density functions with varying shape parameter.

Fig. 2.4. Zoomed portion of Fig. 2.3 showing distinct curve characteristic of Hellinger distance near shape parameter of 2.
2.1.5 Why Hellinger Distance

Hellinger Distance possesses all four properties of a metric and provides the sharpest discriminating abilities between similar probability densities. [7] shows that it also exhibits robustness to outliers while maintaining optimality when used in classification, a pair of properties that is not matched by other distances. This makes a Hellinger distance based classifier robust to outliers without sacrificing performance. [8] shows that minimum Hellinger distance estimators have the smallest sensitivity to certain types of contamination. This makes it ideally suited for situations where data is insufficient or models are imperfect.

2.2 The Minimum Hellinger Distance Classifier

With the structure of a minimum distance classifier and with Hellinger distance as the distance metric of choice, the Minimum Hellinger Distance Classifier (MHDC) can be formed. The MHDC is simply the minimum distance classifier with the Hellinger distance used for forming the score. This thesis seeks to solve a binary depth classification problem, so the two models are called shallow and deep. The data is called the received signal. The received signal and shallow and deep models are represented by probability density functions. They are represented, respectively, as \( p_s(x), p_a(x) \) and \( p_d(x) \) where \( x \) is a parameter vector. With those definitions the MHDC can be drawn as in Fig. 2.5.

![Flow diagram of the minimum Hellinger distance classifier](image)

**Fig. 2.5.** Flow diagram of the minimum Hellinger distance classifier

Under ideal performance of the binary depth classifying MHDC the Hellinger distance between a received signal originating from a deep source and the deep model would be 0 and the distance
between the deep-originating signal and the shallow model would be 1. The opposite would be true for a signal originating from a shallow source. This would correspond to a score, $\Delta$, being 1 for deep sources and -1 for shallow sources, with a discontinuity at the midpoint. Fig. 2.6 shows a plot of ideal score versus received signal depth and illustrates the score for ideal performance.

![Fig. 2.6. Plot of the ideal score in MHDC. A sharp transition at the boundary between the two classes (deep and shallow signal depth) and scores of exactly 1 and -1 show an ideal classification result.](image)

Actual performance shows maximum and minimum score values much closer to each other without a discontinuity, although a trend is visible that differentiates deep and shallow. Fig. 2.7 shows scores generated from acoustic propagation simulations. It is clear that a threshold parameter is required in actual situations, as the decision will not change over at precisely 0. It is also clear from the data crossing the threshold that the correct classification rate will not be one hundred percent. To assess these errors and examine performance, the MHDC is compared to a Bayesian classifier whose performance is widely studied.

### 2.3 Comparison of MHDC to Bayesian Methods

#### 2.3.1 Brief Introduction To Bayesian Methods

The Likelihood Ratio, as introduced in Equation (2.2), forms the basis for a classification scheme called the Log-Likelihood Ratio Classifier (LLRC). The same two models that were used in the MHDC are used to form the likelihood ratio. The likelihood of a received signal is the joint
Fig. 2.7. Plot of actual score in MHDC. A downward trend is seen and a threshold can be drawn to make classification decisions but a 100% classification rate is impossible.

likelihood of all of the points in that signal, i.e.

$$\lambda(x) = \lambda(x_1, x_2, ..., x_n)$$  \hspace{1cm} (2.14)

This likelihood would be very difficult to calculate if the data were not independent. As a simplifying assumption, it is commonly asserted that the data are independent. This means the joint likelihood of the data is simply the product of the likelihood of each point, i.e.

$$\lambda(x) = \lambda(x_1)\lambda(x_2)\cdots\lambda(x_n)$$  \hspace{1cm} (2.15)

It is important to note that this assumption is not, in general, accurate for acoustic time series data. In fact, its accuracy will greatly affect the performance of a classifier based on this method as will be seen later in this thesis. This product of likelihoods can quickly become problematic numerically as the numbers grow, especially if the data set is large. A common practice that alleviates the numeric problems is to sum the log of the likelihoods instead of taking a product of the likelihoods. The summed log-likelihood can then be compared to a threshold to make a binary decision. Fig. 2.8 contains a diagram of the classifier. This method is standard in classification and detection literature and more information can be found in [15] and [16] among others. The log-likelihood ratio is an efficient estimator, meaning that it can achieve the Cramer-Rao bound on the variance of its estimate. Therefore when the received signal comes from one of the two models, meaning the likelihood ratio is a good estimator for the received signal, this method will
give the best classification possible. As it is a well-known and standard classification method, it will be used as a benchmark to compare to the MHDC.

![Flow diagram for the log-likelihood ratio classifier.](image)

**Fig. 2.8.** Flow diagram for the log-likelihood ratio classifier. An assumption of point-to-point independence in the received signal is an important requirement. Model probability density functions are estimated from simulations of models.

### 2.3.2 MHDC vs. LLRC

The MHDC and LLRC can then be compared. The LLRC uses a single likelihood ratio, $\lambda(x)$, formed by the probability density functions for the shallow model, $p_s(x)$, and the deep model, $p_d(x)$. The MHDC uses two likelihood ratios. The first, $\lambda_s(x)$ is formed with the received signal probability density function, $\hat{p}_r(x)$ and the shallow probability density function $p_s(x)$. The second, $\lambda_d(x)$ is formed with the received signal probability density function, $\hat{p}_r(x)$ and the deep probability density function $p_d(x)$. The quantities the classifiers use for classification, sometimes called contrast functions, also differ. The LLRC uses a sum of logarithms of the previously formed likelihood ratio evaluated at the values of the received signal whereas the MHDC takes the expectations of a functional of the ratios with respect to the received probability density function. The two classifiers then compare these results to a threshold in a similar fashion. These differences are shown in Table 2.3.

A common method of comparing Bayesian classifiers is the Receiver Operating Characteristic (ROC) curve.
Table 2.3. Comparison of MHDC and LLRC

<table>
<thead>
<tr>
<th>Use of LR</th>
<th>Likelihood Ratio</th>
<th>Minimum Divergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda(x) = \frac{p_s(x)}{p_d(x)}$</td>
<td>$\lambda_s(x) = \frac{\hat{p}_r(x)}{p_s(x)}$, $\lambda_d(x) = \frac{\hat{p}_r(x)}{p_d(x)}$</td>
<td></td>
</tr>
<tr>
<td>Contrast Functions</td>
<td>$\Delta(r) = E_{\hat{p}<em>r(x)}(\phi(\lambda_s(r_i))) - E</em>{\hat{p}_r(x)}(\phi(\lambda_d(r_i)))$</td>
<td></td>
</tr>
<tr>
<td>Threshold Comparisons</td>
<td>$\lambda_r \geq \eta_{H_s} \quad \Delta(r) \geq \eta_{H_d}$</td>
<td></td>
</tr>
</tbody>
</table>

2.3.2.1 Receiver Operating Characteristic Curves

Given a binary hypothesis test, i.e. a choice between Hypothesis 1 ($H_1$) and Hypothesis 2 ($H_2$), a Receiver Operating Characteristic (ROC) curve is a plot of the Probability of False Alarm ($p_{fa}$) versus the Probability of Detection ($p_d$) for a particular classifier. Fig. 2.9 illustrates the probabilities of choosing $H_1$ or $H_2$ called $p_1$ and $p_2$ respectively.

![Fig. 2.9. A typical pair of probability density functions representing the probability of choosing two different hypotheses](image)

The Probability of False Alarm is defined as the probability that the classifier will choose $H_1$ when $H_2$ is the truth. The Probability of Detection is defined as the probability that the classifier will choose $H_2$ when $H_2$ is the truth. A threshold $\eta$ can be defined such that all probability mass in $p_1$ beyond $\eta$ is equal to $p_{fa}$ or

$$p_{fa} = \text{prob (choose } H_1 \text{ when } H_2 \text{ is true}) = \int_{\eta}^{\infty} p_1(x) \, dx$$

(2.16)
and all probability of $H_2$ beyond $\eta$ is equal to $p_d$ or

$$p_d = \text{prob}(\text{choose}H_2\text{when}H_2\text{is true}) = \int_{\eta}^{\infty} p_2(x) \, dx \quad (2.17)$$

**Fig. 2.10.** Illustration of threshold, $p_d$, and $p_{fa}$ used in calculating the performance of a classifier.

Fig. 2.10 illustrates the probabilities of the hypotheses and their relationships with the threshold and probabilities of detection and false alarm. By sweeping the value of the threshold, $p_{fa}$ and $p_d$ can be calculated across their entire ranges, producing an ROC curve. Fig. 2.11 shows a typical ROC curve.

A particular point along the curve will determine a threshold and operating $p_{fa}$ and $p_d$ for the classifier. There are a variety of methods for choosing a threshold and hence an operating point and those methods are discussed in detail in [15] and [16] but will not be covered here. ROC curves contain some general properties. All ROC curves are concave downward. If the curve is not concave down, a randomized test would do better. The $p_{fa} = p_d$ line represents a “coin-flip” test. A ROC curve should never be below the $p_{fa} = p_d$ line. If it is, the decision should simply be reversed and the curve will once again be above that line. Other properties also exist, but are not relevant to this discussion.

### 2.3.3 ROC Curves Based On Gaussians

The processes for making ROC curves from Gaussian data using the LLRC and MHDC are similar. Both classifiers begin with two probability density functions, $p_1$ and $p_2$, representing
Fig. 2.11. A typical ROC curve along with the "chance line". Performance of a classifier correlates with the area under the ROC curve.

the probability of choosing hypothesis $H_1$ and $H_2$, respectively. In the present example, both probability density functions have zero mean. $p_1$ has unity variance while $p_2$ has a variance of 1.2. Fig. 2.12 shows these two probability densities plotted on the same axes. Realizations of length $N$ are formed from $p_1$ and $p_2$, respectively named $x_1$ and $x_2$. Fig. 2.13 contains a typical illustration of the pair of realizations. To set up the LLRC the log-likelihood ratio must be calculated. Fig. 2.14 shows a typical log-likelihood function for the case of two Gaussians with zero mean and different variance formed from the analytical probability density functions $p_1$ and $p_2$. To set up the MHDC probability density estimates $\hat{p}_1$ and $\hat{p}_2$ must be made for the two realizations $x_1$ and $x_2$, respectively. Fig. 2.15 shows typical probability density function estimates derived from $x_1$ and $x_2$. To form a ROC curve with $M$ points, $M$ realizations must be created and tested on the classifier, yielding $M$ detection statistics. Fig. 2.16 and Fig. 2.17 show typical detection statistic and score results for the LLRC and MHDC, respectively. The detection statistics of the LLRC are on a different scale than the detection statistics of the MHDC. The absolute value of the detection statistic or score is not important in classification. The relative difference between the detection statistics or scores for each hypothesis are what determine classification performance. Density estimations are then performed on the detection statistics as shown in Fig. 2.18 and Fig. 2.19. Using the threshold integration methods described in equations 2.16 and 2.17 the detection statistic densities may be turned into ROC curves. Fig. 2.20 and Fig. 2.21 show typical ROC curves for the LLRC and MHDC, respectively.
Fig. 2.12. Underlying densities for Gaussian test data. Each probability density function represents the probability of choosing the corresponding hypothesis based on the value of $x$.

Fig. 2.13. Realizations of $p_1$ and $p_2$ called $x_1$ and $x_2$, respectively. These are typical realizations of a Gaussian random variable. The probability density functions used to generate these realizations are plotted in Fig. 2.12.
Fig. 2.14. Log-likelihood ratio function calculated from $p_1$ and $p_2$.

Fig. 2.15. Probability density functions $\hat{p}_1$ and $\hat{p}_2$ estimated from $x_1$ and $x_2$ using kernel density estimates for use as MHDC models. With infinite data and a perfect density estimation these two curve would look exactly like $p_1$ and $p_2$. 
Fig. 2.16. Detection statistic results of a Monte Carlo simulation of the LLRC.

Fig. 2.17. Score results of a Monte Carlo simulation of the MHDC.
Fig. 2.18. Probability density functions for detection statistics for the LLRC. These are the curves that will be thresholded and integrated for creation of a ROC curve.

Fig. 2.19. Estimated probability density functions for detection statistics for the MHDC. These are the curves that will be thresholded and integrated for creation of a ROC curve.
Fig. 2.20. A representative ROC curve for the LLRC’s performance on Gaussian data.

Fig. 2.21. A representative ROC curve for the MHDC’s performance on Gaussian data.
2.4 Performance On Gaussian Data

2.4.1 Ideal Performance

According to the Neyman-Pearson Lemma, the log-likelihood ratio classifier is optimal for classification when priors are known [15]. The minimum Hellinger distance classifier also exhibits this property, although asymptotically as described in [7].

Indeed as the number of points used in classification increases the performance of the MHDC becomes closer to that of the LLRC. It is therefore important to realize that the performance of the MHDC on a limited amount of data will not equal that of the LLRC but this inequality is diminished by a larger data set. Fig. 2.22 shows the performance of the classifiers for increasing realization size. Although the curves are similar, they become closer together as realization size increases, reflecting the asymptotic optimality of the MHDC. Realization sizes are 1,000, 5,000, 10,000 and 20,000, each with 1,000 Monte Carlo runs.

![ROC curves showing the effect of increasing realization size on the performance of the LLRC and MHDC. Realization size increases toward the upper left corner of the plot.](image)

Fig. 2.22. ROC curves showing the effect of increasing realization size on the performance of the LLRC and MHDC. Realization size increases toward the upper left corner of the plot.

Fig. 2.23 shows the performance of the classifiers with changing number of Monte Carlo runs. These curves represent trials with 500, 1,000, 5,000 and 10,000 Monte Carlo runs. All of the curves lie on top of each other or nearly so, showing that a number of runs beyond 500 is not necessary.

Fig. 2.24 shows the performance of the LLRC and MHDC with changing variance in the source distributions. In these experiments the realizations are 10,000 points long and 1,000 runs are simulated. Each curve shows the performance of the classifier where the two models are...
Fig. 2.23. ROC curves showing performance of the MHDC and LLRC for a varying number of Monte Carlo runs. There are 4 curves for each classifier.

a unity variance zero mean Gaussian and a zero mean Gaussian with slightly larger variance. Starting with the bottom rightmost curve with a variance of 1.0 versus 1.01 and moving to the upper left corner the variance difference increases by 0.01 every curve up to a variance difference of 1.0 versus 1.05.

2.4.2 Contamination Models

2.4.2.1 Model Types

Another aspect of the classifiers to be examined is their performance in the presence of data contamination. Although not exactly the same, contamination emulates a mismatch between data and model. A standard model of contamination is the epsilon contamination model in which the probability density function of the received data $f'$ ($x$) is a convex combination of a true density $f$ ($x$) and a contamination density $g$ ($x$), where epsilon is called the contamination factor:

$$f'(x) = (1 - \epsilon) f(x) + \epsilon g(x)$$  \hspace{1cm} (2.18)

An epsilon of 0 corresponds to an uncontaminated density while an epsilon of 1 corresponds to a received data set made up entirely of contamination. A second contamination model characterizes the received data probability density function as varying from $f$ ($x$) to $f$ ($x$) + $g$ ($x$):
Fig. 2.24. ROC curves showing the effect of changing variance on the performance of the LLRC and MHDC. The difference in variance for $p_1$ and $p_2$ increases toward the upper left corner.

$$f'(x) = (1 - \epsilon) f(x) + \epsilon \left[ \frac{f(x) + g(x)}{2} \right]$$

(2.19)

2.4.2.2 Contamination Types

There are two different types of contamination, inliers and outliers. Inliers consist of probability mass that is statistically likely to come from the true probability density function whereas outliers come from mass that is statistically unlikely to have come from the true probability density function. While this is a qualitative definition and probability mass can ambiguously belong to either type, the experiments performed in this thesis use contamination that is strongly an inlier or outlier. Figure 2.25 shows a cartoon of a gaussian probability density function with inlier and outlier contamination. The locations of the delta functions corresponding to inlier and outlier mass are respectively 1.7 and 5 in this cartoon.

2.4.2.3 Model Implementations

A realization of $f'(x)$ with length $N$ is desired. The simplest method of simulating the first type of contamination model would be to produce a data set consisting of $(1 - \epsilon) N$ samples from $f(x)$ and $\epsilon N$ samples from $g(x)$. While the individual samples will not have been drawn directly from $f'(x)$, the statistics of the realization will be consistent with $f'(x)$. This method presents several problems. If $g(x)$ consists of outliers, the MHDC will see an effective sample size of $(1 - \epsilon) N$ when the Hellinger distance between $f(x)$ and $f'(x)$ is calculated and its performance
will reflect the decrease in sample size. The LLRC will have one of two problems. The outlying contamination mass can contribute likelihood to the correct hypothesis or the incorrect hypothesis simply due to the construction of the likelihood ratio. The LLRC’s performance will therefore be inflated or deflated arbitrarily. The MHDC’s performance is robust to this effect.

A second method of producing a realization of $f'(x)$ is to draw $N$ samples from $f(x)$ and $\epsilon N$ samples from $g(x)$ to form a sample of actual length $(1 + \epsilon)N$. In the case of outlying contamination mass the MHDC will see an effective samples size of $N$. The results of the LLRC using this method will have the same arbitrary inflation or deflation of likelihood as the other method. In the case of inlying contamination mass the MHDC and LLRC will both see the full sample length of $(1 + \epsilon)N$, slightly raising performance but raising it equally for the MHDC and LLRC.

A second consideration in performing tests on the effects of contamination on the classifiers is where contamination is applied. If contamination is applied to both $p_1$ and $p_2$ the detection statistics are equally affected and will therefore not yield a difference in performance when the ROC curve is calculated. The contamination should therefore only be added to one of the probability density functions.

Fig. 2.25. Cartoon demonstrating inliers and outliers
Table 2.4. Means of contamination statistic probability density functions as results of the LLRC for different contamination methods and the change in mean from uncontaminated to contaminated probability density functions as a percentage of the original mean are displayed. Contamination is specified in the first column by which probability density is contaminated and the location of the probability mass corresponding to contamination.

| Contamination | LLRC | | MHDC | | | |
|---------------|------|----------|----------|----------|----------|
|                | $\mu_1$ | $\mu_2$ | $\Delta \mu_1$ (%) | $\Delta \mu_2$ (%) | $\mu_1$ (x10$^{-3}$) | $\mu_2$ (x10$^{-3}$) | $\Delta \mu_1$ (%) | $\Delta \mu_2$ (%) |
| None          | 2.03  | -1.97   | 0         | 0         | 1.60      | -1.51      | 0               | 0               |
| $p_1$ and $p_2$ at -6 | 3.80  | -0.15   | 87        | 92        | 0.77      | -0.79      | -52             | 48              |
| $p_1$ and $p_2$ at 6    | 0.28  | -3.75   | -86       | -90       | 0.79      | -0.74      | -51             | 51              |
| $p_1$ at -6          | 3.78  | -2.07   | -84       | -5        | 0.78      | -1.51      | -51             | 0               |
| $p_1$ at 6           | 0.32  | -2.08   | -84       | -5        | 0.77      | -1.57      | -52             | -4              |
| $p_1$ at -1          | 2.27  | -1.94   | 12        | 2         | 1.80      | -1.54      | 13              | 2               |
| $p_1$ at 1           | 1.75  | -2.06   | -14       | 5         | 1.25      | -1.52      | -22             | 1               |

2.4.3 Contamination Results

2.4.3.1 Numerical Results

Simulations with different combinations of contamination applied to different distributions at different locations were run and the results are summarized in Table 2.4. Analysis of these results follows. In all of these simulations the second method of adding contamination, where $\epsilon N$ points are added to the end of a realization of length $N$ was used. To ensure that inliers and outliers are strongly classified as such, mass locations of 1, -1, 6 and -6 are used. $\mu$ is the mean of the detection statistic probability density function with the contamination type specified in the first column. $\Delta \mu$ is the difference in mean between the probability density functions for the uncontaminated detection statistics and the contaminated detection statistics. Subscripts indicate which of the two hypotheses were tested.

The first two contaminated simulations put outlier mass on both $p_1$ and $p_2$ at 6 and -6. The detection statistics for the LLRC see approximately equal changes in mean of about 90% from the nominal simulation for both $\mu_1$ and $\mu_2$. The scores for the MHDC separate with $\mu_2$ increasing by 50% and $\mu_1$ decreasing by 50% from nominal.

The next set of contaminated simulations shows the effect of outlier mass when applied only to $p_1$. This shows that contamination of only one realization will result in changes of only the detection statistic associated with it, i.e. contamination on $p_1$ only affects $\mu_1$. The sign of $\Delta \mu$ will determine if the two detection statistic probability density functions get closer or further apart. For the case of contamination mass applied to $p_1$ at -6, the difference between $\mu_1$ and $\mu_2$ for the LLRC is 5.51 whereas the difference in the case of contamination mass at 6 is 2.46. The difference in detection statistic means directly correlates to performance, indicating that the LLRC will perform significantly better with contamination at -6 and significantly worse with contamination at 6. This means the location of contamination can arbitrarily drive the performance of the LLRC up or down depending on where it lies. The difference between the means in the case of the MHDC does not change significantly, indicating it is more robust to the
location of the contamination.

Contamination by inliers either at 1 or -1 shows a similar change in mean for both the MHDC and LLRC, indicating a lack of robustness to inliers in both classifiers, as expected.

2.4.3.2 Graphical Results

Simulations of contamination models demonstrate the robustness of the MHDC to outliers. To simulate outliers, contamination mass at x=6 is added to a realization of a unity variance, zero mean Gaussian. To simulate inliers, contamination mass at x=1 is added to a realization of a unity variance, zero mean Gaussian. The mass at these contamination sites is varied, producing a series of a ROC curves. Fig. 2.26 shows the performance of the LLRC and MHDC classifiers on data contaminated with inliers. The presence of these inliers does not significantly affect either classifier although a slight decrease in performance of the MHDC with increasing contamination due to the decrease in effective sample size is visible.

![ROC curves](image)

**Fig. 2.26.** Performance of the MHDC and LLRC with inliers at x=1. Contamination levels are 0, 0.05, 0.10, 0.15, 0.20 and 0.25 percent. There is little difference in performance between the classifiers.

Fig. 2.27 shows the performance of the MHDC and LLRC in the presence of outliers. The MHDC’s performance is slightly degraded by these outliers, but the LLRC’s performance is severely impaired. It is important to note that the LLRC’s performance would be boosted if the contamination were applied to the probability density representing the other hypothesis. This is a demonstration of the robustness of the MHDC to outliers, a property that is beneficial when there is data-model mismatch in a system.

This experiment highlights the advantage of the MHDC over the LLRC for data containing
outliers while demonstrating parity for data containing inliers.

### 2.5 MHDC With Density Estimation

Armed with acoustic simulations, some pre-processing and density estimation tools, the complete MHDC can be constructed. Density estimation methods are covered in Appendix A. A kernel density estimate of the received signal forms the model \( p_r(x) \) that is used in distance calculation along with the models \( \hat{p}_s(x) \) and \( \hat{p}_d(x) \) derived from acoustic propagation code as described earlier. Fig. 2.28 shows a complete flow diagram of the MHDC with density estimation included.
Fig. 2.28. Flow diagram for the full MHDC with probability density function estimation
Chapter 3

Application to Synthetic Acoustic Data

3.1 Introduction

With the MHDC defined and its performance on Gaussian distributed synthetic data quantified and compared to the LLRC, it is important to look at performance on underwater acoustic data. To test the classifiers, the received data and models must be defined. A Monte Carlo simulation is performed, varying several environmental parameters, to obtain a statistical picture of acoustic propagation through the environment. A simulation of the nominal environment is considered to be the received data and the statistics from the Monte Carlo simulation form the class models.

3.2 Acoustic Simulations

First it is necessary to generate the acoustic data. Underwater acoustic propagation simulations require two things; a propagation code and an environment. The chosen propagation code is called RAM and the chosen environment is a variation on the GENLMIS benchmark environment, both defined and explained below.

3.2.1 The Propagation Code - RAM

There are a variety of methods of simulating underwater acoustic propagation. Popular methods include ray tracing, normal modes, wavenumber integration, finite elements and the parabolic equation (PE). For this work the PE method will be used in simulation. A popular implementation of this method is RAM (Range Acoustic Model), a code developed by Mike Collins which has become standard for this propagation method. The details of its implementation and the PE methods can be found in [17] and [18]. RAM is a grid-based code and therefore requires the
definition of a range and depth step that are constant throughout the simulation. The code also
takes as its input a starting field and environmental parameters. The product of the code is a
calculated pressure field on the pre-defined grid of range and depth. The convention of depth
increasing from 0m at the surface and range increasing from 0m at the source is used. With
knowledge of the initial field and the pressure field produced, a field of transmission loss in deci-
bels can be defined. This transmission loss field is the source of all of the synthetic acoustic data
processed.

### 3.2.2 The Acoustic Environment

The ocean can be thought of as having two general portions, the water and the sediment. The
propagation of sound underwater depends heavily on the properties of the water and the sediment.
For these simulations the surface is a flat pressure release boundary, meaning it is perfectly
reflective. The properties of the water and sediment then make up the possible variability in the
environmental model. The characteristics of the water are summarized in 2 parameters: sound
speed profile and bathymetry. The simulations for this work define a flat sea floor but vary the
water column depth when variability is desired. Variability in the sound speed profile is produced
by keeping the sound speed at the surface constant while changing the sound speed at the bottom
of the water column. The sediment is characterized by three parameters: sound speed profile,
attenuation and density. All three of these parameters are constants in this work.

The specifics of the environment used for these simulations are adapted from a Naval Research
Laboratory sponsored workshop on matched field processing. [19] describes the workshop, which
included a set of problems that were used to test different matched-field algorithms. The nominal
environment chosen for this work is a modification of the true environment for GENLMIS. Fig.
3.1, adapted from the report, shows the parameters that define the environment. In this plot, and
following the convention, z is 0m at the surface and increases into the water, so the bottom is at
a positive z of 100m. c(z) represents sound speed in m/s both in the water and in the sediment.
Sound speed profiles are linear interpolations between the specified points. The density and
attenuation of the sediment is constant. For simulation purposes a highly attenuating layer is
present at a depth much greater than the sea bottom to facilitate computation, a practice which
is standard when using RAM for simulation. This allows the energy that would radiate into the
earth through the bottom to leave the simulation gracefully.

#### 3.2.2.1 The Nominal Environment

This environment is the underlying environment for the synthetic data. Variations of this envi-
ronment are processed to form the models used in classification and the received data used in
classification comes from this environment. The nominal environment has a water column depth
of 100 meters and a linear sound speed profile that changes from 1500 m/s at the surface to 1490
m/s at the bottom of the water in a linear fashion. The parameters describing the sediment
create a reflective bottom, allowing long range acoustic propagation, a feature that is beneficial
3.2.2.2 Variations Of The Environment

A Monte Carlo simulation of the environment is required to capture statistical uncertainty in environmental parameters. This is accomplished by simulating variations on two environmental parameters that are assumed to be uncertain. The water column depth has a uniform distribution between 95 and 105 meters, sampled at 0.5 meter increments. There are 21 samples of water column depth with the nominal environment in the middle of the range. The sound speed profile is varied by changing the sound speed at the bottom of the water column. It varies from 1480 m/s to 1500 m/s with a step size of 1 m/s, creating profiles that range from strongly downward refracting to iso-velocity and leaving the nominal environment in the center. Fig. 3.2 illustrates this variation.

3.2.2.3 Source And Receiver Characteristics

The source simulated is an acoustic point source emitting a 100Hz tone. It is located at 7.2m depth as specified in the GENLMIS experiment. It is assumed the principle of reciprocity holds, allowing the source and receiver to be interchanged, effectively making the receiver fixed while the source is at different ranges. This assumption greatly simplifies the simulations and is approximately true given small source-receiver relative velocity. These experiments assume the source and receiver are at constant depth.
3.2.3 Pre-processing of RAM’s Output

The output of RAM is a transmission loss field that includes propagation in the sediment. Fig. 3.3 shows the raw transmission loss output of the RAM simulation.

The first step in processing is to ignore the transmission loss field in the sediment, as a source is not going to be there. The next step is to separate the far-field propagation from the near-field propagation. It is assumed that sources will be of sufficient distance from the receiver that only far-field propagation is of interest. In these simulations a range of 1000m and further is considered far-field. This is accomplished by simply ignoring the transmission loss field at short range. These two steps create a transmission loss area of interest and Fig. 3.4 shows the RAM simulation cut down only to that area of interest.

The transmission loss field produced by RAM has a $r^{-\frac{1}{2}}$ range dependence due to spreading of the acoustic wave as can be seen in Fig. 3.4 above. As it is assumed that source range and level are unknown, this dependence must be corrected to maintain the proper ambiguity in the statistical models. The spreading-corrected transmission loss field can be seen in Fig. 3.5.

This transmission loss field is now prepared for processing. Peaks and troughs have approximately even magnitude across range. This field will be used both to create models and as received data for simulation of the classifiers.

3.3 Received Data

The object of the classification task is to decide whether a received signal originates from a source that is deep in the water channel or shallow. It is therefore necessary to construct a received data set for each case. The shallow received signal is taken to be the transmission loss in the nominal
Fig. 3.3. Transmission Loss (TL) as a function of depth and range is the output of RAM. TL is plotted as a decibel quantity.

Fig. 3.4. Transmission Loss in area of interest
Fig. 3.5. TL is plotted with ranged-based spreading corrected by subtracted an assumed cylindrical spreading factor.

environment at 30 meters depth across the simulated range. The deep received signal is taken to be the transmission loss in the nominal environment at 70 meters depth across the simulated range. Both of these signals have the same record length which is equal to the range covered divided by the range step from the acoustic simulation. The area of interest in the simulated environments has 10,000m of range with a 1m range step or 10,000 points. Fig. 3.6 illustrates the data chosen for each received data set.

To test the classifiers ROC curves are created. Each point in the ROC curve represents the performance of the classifier on a received data set. If the entire 10,000 point set is used as a single data set, only one point is available for a ROC curve. The received data set must therefore be divided into smaller portions and treated as N data sets to get N points on the ROC curve. This division of data comes at a cost. The statistics of the entire data set must be represented by each of the subsets. It is therefore prudent to investigate several different ways to divide up the entire data set. In each of these methods \( r_i \) is defined as the indices of data set i, where

\[
i = [1, 2, ..., N]
\]

and the length of the entire data set is L. The length of a single sample is then \( \frac{L}{N} \).

3.3.1 Sequential Samples

The first division method is to separate the data sequentially, similar to the Bartlett method in spectrum estimation. In this case the indices follow the form
which produces $N$ equal size portions of the whole data set. The problem with this method is that there is significant spatial correlation in the data. This sequentially sampled set therefore does not represent the statistics of the entire data set. As the classification methods to be tested are based on the statistics of the data, this is an undesirable sampling scheme.

### 3.3.2 Interleaved Sequential Samples

A second way to divide the data is to sample in an interleaved fashion. In this case the indices follow the form

$$r_i = i + [0, N, 2N, ..., L - N]$$

where it is assumed that $L$ is a multiple of $N$. This method also produces $N$ equal sized portions of the whole data set. It has the added benefit of creating an uncorrelated data set as long as the correlation length of the data is not $N$, a situation which is not likely. This method is also used in spectrum estimation. The lack of correlation and distributed sampling yields data subsets that have similar statistics to the entire data set.
3.3.3 Uniform Random Samples

A third way to divide the data is to sample randomly. In this case the indices are randomly chosen according to a uniform distribution. This will produce \( N \) equal sized portions of the data set with data chosen randomly from the whole set. It also has the benefit of uncorrelated data but has the drawback that some data can be double counted while other data is not represented at all. These indices can be written

\[
    r_i \sim U(1, 10000) \in \mathbb{N}
\]

This method is related to but not equivalent to drawing random samples from the distribution of the entire data set, creating subsets purely statistically.

3.3.4 Results of Sampling Study

To test which sampling method produces the most accurate statistical representation of the entire data set, ROC curves were generated using the MHDC, models derived from the entire data set, and received data generated from the three different sampling methods. As the models and received data come from the same data, perfect performance is possible. It is found that interleaved sampling of the entire received data set produces the best results while sequential sampling produces the worst. As predicted the performance of the methods varies with \( N \). Fig. 3.7 represents \( N=14 \) and is representative of the performance differences. The curve representing the performance of the periodic method is hard to see in the plot because \( p_d=1 \) for all \( p_{fa} \) in that situation, corresponding to perfect performance.

From this point on it is assumed that data sets from the synthetic data are drawn using the interleaved sampling method as it provides the best representation of the statistics of the data set and hence performance. However, even with this sampling method the probability density functions produced from the smaller data sets are not ideal. The larger \( N \) becomes, the less the probability density function resembles that of the full data set. This effect can be seen in Fig. 3.8, Fig. 3.9, and Fig. 3.10. Each of these figures shows a plot of the probability density functions from the entire data set in red and the probability density functions from each smaller data set in blue. Only the data used in the deep model is included in these diagrams, as the data from the shallow model shows similar results.

The shape of the class model probability density functions are very important to the performance of the MHDC because its detection statistic is heavily influenced by point-wise differences in the probability density functions of the class models and data. The LLRC, on the other hand, is dependent on the general location of the probability mass because its detection statistic is independent of the probability density function of the data, being derived from the log-likelihood ratio function, which is formed solely from the class models. The performance implications of this distinction will become evident in Section 3.5.
Fig. 3.7. Results of data sampling experiments

Fig. 3.8. PDF of TL with N=10 subsections in blue and N=1 subsection in red
Fig. 3.9. PDF of TL with $N=100$ subsections in blue and $N=1$ subsection in red

Fig. 3.10. PDF of TL with $N=1000$ subsections in blue and $N=1$ subsection in red
3.4 Models

The classifiers must have statistical models to operate. Each classifier needs a pair of models; one for a shallow source and one for a deep source. To test the generalization capabilities of the classifiers four different pairs of models are formed representing four different levels of knowledge regarding the source.

3.4.1 Construction of Models

Models are constructed by forming probability density functions of portions of the transmission loss field using a kernel density estimate. In every case to be tested both a deep and a shallow method are created. Different levels of knowledge may be simulated by including different portions of data from simulations. In all cases the portions of data included in the model are simply grouped together and used as the data fed into the kernel density estimator. There are two different types of knowledge incorporated in the models; knowledge of the source depth and knowledge of the environment. For each type two states of knowledge are assumed; one where the situation is known, and one where the situation is uncertain. Knowledge of the environment is simulated by using simulations only from the nominal environment to produce the model. Uncertainty in the environment is represented by incorporating all 441 different environments representing the uniform uncertainty in sound speed profile and water column depth in the model. Knowledge of the source depth is represented by the source depths included in the model. Uncertainty is represented by incorporating a window of source depths whereas a strong knowledge is represented by incorporating a single source depth. The uncertain source depth models use a window of 25 to 35 meters for the shallow model and 65 to 75 meters for the deep model while the known source depth models use the data from a depth of 30 meters for the shallow model and 70 meters for the deep model. Fig. 3.11 shows the window used for the known depth situation while Fig. 3.12 shows the windows used for the uncertain depth situation.

The four combinations comprising the two different states of the two different types of knowledge yield four pairs of models. Fig. 3.13 shows the resulting probability density functions for the four different situations.

The problem clearly becomes more difficult as knowledge is removed and the probability densities for deep and shallow become more similar. This can be seen by comparing parts (b) and (d) or parts (a) and (c) from Fig. 3.13. It is also seen that knowledge of depth does not appear to be as important as knowledge of environment in shaping the probability densities, as can be seen by comparing Fig. 3.13 parts (a) and (b) or parts (c) and (d).

3.5 Results

To evaluate the classifiers ROC curves are created. Each of the plots in Fig. 3.14, Fig. 3.16, Fig. 3.15, and Fig. 3.17 contains a series of ROC curves for N=100, 200, 300, 400, 500 in blue for the LLRC and green for the MHDC with the chance line in black. In general there is an
Fig. 3.11. TL data included in the known depth situation

Fig. 3.12. TL data included in the uncertain depth situation
Fig. 3.13. Models produced for different levels of knowledge of the source and environment

inverse correlation between performance and N as predicted. The juxtaposition of the LLRC and MHDC shows a difference between the performances of the classifiers with different knowledge levels.

3.5.1 Known Depth, Known Environment

The set of curves in Fig. 3.14 shows the known depth and known environment situation. This is the least complex situation and indeed the data used for creating the models is exactly the same as the data used for testing. The performance of the LLRC is almost perfect, as would be expected. The performance of the MHDC suffers for large N because the probability densities produced from smaller data sets are not accurate reproductions of the statistics of the entire data set as can be seen in Figs. 3.8, 3.9 and 3.10. As previously stated, the LLRC is not affected by this because its performance is not directly dependent on the estimated probability density function of the data subset.

3.5.2 Unknown Depth, Known Environment

The set of curves in Fig. 3.15 shows the unknown depth and known environment situation. There is more uncertainty in this situation than the first and the performance of both classifiers is, as expected, worse. While the performance of both classifiers has decreased, the LLRC has suffered more degradation than the MHDC. The degraded performance is due to the increased similarity between deep and shallow models when the added ambiguity of unknown depth is
Fig. 3.14. Results for known depth and known environment

included, as can be seen in Fig. 3.13. As the class models become more similar, the magnitude of the log-likelihood function decreases. As the LLRC depends upon summed samples from the log-likelihood function for classification, its ability to differentiate between classes suffers. Although the MHDC also suffers due to the similarity in probability density functions, the nature of the Hellinger distance as an integration of the probability density functions instead of a pointwise evaluation of a function helps the MHDC maintain classification abilities.

3.5.3 Known Depth, Unknown Environment

The set of curves in Fig. 3.16 shows the known depth and unknown environment situation. Performance is similar to the known depth, known environment situation, with both classifiers showing mild decreases in performance compared to that set. This indicates that knowledge of environment, as represented in this experiment, is not important in the classification decision.

3.5.4 Unknown Depth, Unknown Environment

The final set of curves in Fig. 3.17 shows the unknown depth and unknown environment case. As before the performance of the LLRC is decreased by the ambiguity in depth for the same reasons as in the known environment case. The results echo those of the unknown depth and known environment situation, again displaying the robustness of the MHDC to ambiguities in statistical class models.
Fig. 3.15. Results for unknown depth and known environment

Fig. 3.16. Results for known depth and unknown environment
3.6 Conclusions

The Minimum Hellinger Distance Classifier is shown to have robustness in classifying the depth of synthetic acoustic data in the presence of uncertainty in depth and environmental parameters while maintaining optimal performance when those parameters are known. This robustness coupled with nominal optimality makes a strong case for the MHDC as a useful classifier in underwater acoustics. While it is found that knowledge of the depth of a source is more important than knowledge of the acoustic environment for this experiment, it is not possible to generalize this finding without significant further study.
Chapter 4

Application to Real Acoustic Data

4.1 Introduction

With the attributes of the MHDC demonstrated on Gaussian data and synthetic acoustic data attention is turned toward its performance on real acoustic data. The particular data set to be analyzed is first presented, along with the approximate acoustic model used in the experiment. The processing steps involved in the experiment are described, then results are presented with the conclusions drawn.

4.2 The SwellEx-96 Experiment

4.2.1 Setup for SwellEx-96

SwellEx-96 was an underwater acoustic experiment located off the coast of San Diego. It involved a variety of underwater acoustic tests but this work focuses on Event S5 which involved a bottomed horizontal line array (HLA South) at 200 m depth and towed tonal sources at two different depths moving at 5 knots. Fig. 4.1 shows the track of the towed sources in blue and the location of the bottomed array with a green star. For this processing the data is restricted to the time period where the source is opening in range with respect to the array. This restriction puts the track on a constant depth contour, allowing a simple environmental model. Eighteen tonal signals were broadcast at frequencies from 100 to 400 Hz at depths of 9 m and 54 m. Each tone at 9 m was paired with a tone at 54 m. The shallower tone broadcast frequencies were 3 Hz below that of the paired deep tone. Irregularities were seen in the data from the highest frequency pair, leaving data from 8 pairs of tones usable in this processing. The deep tones were at frequencies of 112, 130, 148, 166, 201, 235, 283 and 338 Hz, with shallow tones at frequencies 3 Hz lower. A description of the approximate acoustic model used in the simulations follows with an outline of the necessary processing steps following.
4.2.2 Acoustic Model

Sound propagation through the environment was simulated in RAM using a simplified geometry employing a flat bottom, two sediment layers and a four line segment sound speed profile. This model is an approximation of the actual acoustic conditions in the ocean. The sound speed profile has a thin 5m layer of constant sound speed at the surface followed by two transitional layers leading to an iso-velocity layer from 100m to 200m in water depth. The water column depth is constant at 200m. The sediment consists of a 20m sand layer with a much harder rock layer beneath. Fig. 4.2 shows this environment including the density and attenuation parameters for the sediment. The model does not take into account any range- or time-dependent features of that segment of the ocean. While the model simple, it suffices for this purpose.

RAM was run for each of the source frequencies with this environment and 16 transmission loss fields were produced. Statistical models were formed using windows of transmission loss data from a range of 700-7200 m and depths of 5-15 m for the shallow sources and 40-60m for the deep sources. Kernel density estimates were used to form these models in a similar process to the formation of the models for the synthetic acoustic data situation. Fig. 4.3 shows a typical RAM result with the data used highlighted in red. This process yields 16 statistical models which can be divided into 8 pairs of tones at similar frequencies. These models can then be used to form the MHDC and LLRC for testing of the pre-processed Swellex-96 data.
Fig. 4.2. SwellEx-96 simplified environmental model

Fig. 4.3. Sample RAM TL output for SwellEx-96 simplified environment model
4.3 Processing

4.3.1 Pre-processing Of Received Signals

There are eight pairs of data sets, each consisting of a shallow tone and a deep tone and eight pairs of models that correspond to the data sets. For each tone the received data was basebanded using a multiplication by a complex exponential and a low-pass FIR filter with a bandwidth of 2Hz was used to reduce broadband noise and reject all tones at other frequencies, leaving just the complex envelope of the desired signal. As the sampling rate of the original data was much higher than the rate at which the signals varied it contained a large number of identical data points and point to point correlation. The data set was decimated by a factor of 3 then by 10 to reduce its size and make it manageable. The correlation in the signal means that this decimation reduces the number of data points without reducing the information content. The decimation does not remove all correlation in the signal but does reduce it. The correlation could be completely removed by whitening the data, but that would not be feasible for a real-time system. Finally, using GPS information and time stamps from the experiment the data was then projected onto an even range grid. At the end of the pre-processing each data set consisted of about 3.2 million points covering 50 minutes of data evenly spaced in range with a range step of about 2mm.

For processing each set was divided into N segments, contributing N points to the ROC curve. Therefore for N=2 the smaller data sets each have 25 minutes of data and for N=100 about 30 seconds. As with the synthetic acoustic data, the division may be done sequentially or in an interleaved fashion. The interleaved method, as described in Section 3.3.2, produces small data sets with many independent points, leading to a 100% classification rate for small values of N.

Statistical models are then created in a process similar to Section 3.4. The model can be updated in time either as an addition of points included in the density estimation or if a fixed number of points is desired, through a first-in-first-out window.

At this point there are 2 statistical models from simulations and N statistical models from data. Combined, these models and data can run through the MHDC and LLRC, producing sets of detection statistics.

4.3.2 Detection Statistics And ROC Curves

The 8 pairs of data sets each produce N detection statistics. The detection statistics for N=500 are plotted together in Fig. 4.4 and Fig. 4.5 for the MHDC and LLRC, respectively. Both plots show distinct separation between deep and shallow statistics. Another affect that can be seen in these figures is differences in the 8 frequency pairs. Each pair has a different mean for deep and shallow and a different separation between the two. This is caused by having different models for each of the frequency pairs. It will cause problems when trying to combine the data from all 8 data sets into one summary of performance. Fig. 4.6 and Fig. 4.7 in strong contrast to the previous figures show the detection statistics for sequential sampling for MHDC and LLRC, respectively. The sequential sampling scheme leads to detection statistics that show little difference between
deep and shallow data and indeed little difference between the 8 frequency pairs. Although these figures only show the situation for \( N = 500 \), as \( N \) decreases (and sample size increases) the detection statistics begin to separate as could be expected when more independent points are included. Fig. 4.8 and Fig. 4.9 show ROC curves for the sequential and interleaved sampling methods. The visible separation of detection statistics in the interleaved method yields perfect performance in the corresponding ROC curves, whereas the non-separated detection statistics of the sequential method yields poorer performance. This is again a consequence of the number of independent points involved in the calculation.

![Graph](image)

Fig. 4.4. MHDC detection statistics with interleaved sampling (\( N = 500 \))

### 4.4 Results and Conclusions

To better understand the strength of the classifiers a situation between the perfect performance of the interleaved sampling and the mediocre performance of the sequential sampling is desired. To examine a situation between the two the sequential sampling method with small \( N \) will be used. This will produce too few points for a demonstrative ROC curve for each frequency set so the detection statistics for all 8 sets will be combined. This test is not a formal ROC curve because it combines the results of 8 different tests but it can be called an Operating Characteristic (OC) Curve. [20] discusses some of the characteristics of OC curves and compares them to ROC curves. An OC curve can capture the trends and characteristics of a classifier while not being as strict as an ROC. Fig. 4.10, Fig. 4.11 and Fig. 4.12 show OC curves for several different values of \( N \). It can be seen that a smaller \( N \) corresponds to better performance. This is the expected result as
Fig. 4.5. LLRC detection statistics with interleaved sampling (N=500)

Fig. 4.6. MHDC detection statistics with sequential sampling (N=500)
Fig. 4.7. LLRC detection statistics with sequential sampling (N=500)

Fig. 4.8. ROC curves for 8 data sets using LLRC with sequential sampling (N=500). MHDC and LLRC performances are similar.
Fig. 4.9. ROC curves for 8 data sets using LLRC with interleaved sampling (N=500). MHDC and LLRC performances are the same. The ROC curves are hard to see because \( p_d = 1 \) for all \( p_{fa} \) with this sampling method.

A smaller \( N \) corresponds to a larger fraction of the data set being included in each classification. The value of \( N \) here is always a multiple of 8 because it represents the division of the total data. Each of the 8 pairs is broken into \( N/8 \) sequential sections, i.e. \( N=16 \) corresponds to each of the 8 data sets broken into 2 sections. One would expect based on the classification results for each separate pair to get perfect classification performance when \( N=8 \). Perfect performance is not seen, however, due to the mixing of the detection statistics for the 8 pairs. This separation of the 8 sets is visible in Fig. 4.5 and Fig. 4.4 for the LLRC and MHDC, respectively.

The performance of the MHDC and LLRC are shown to be similar for this experiment.
Fig. 4.10. LLRC OC curves with combined data sets. N denotes number of sections the data is divided into.

Fig. 4.11. MHDC OC curves with combined data sets. N denotes number of sections the data is divided into.
Fig. 4.12. MHDC and LLRC performance on SwellEx data set
Conclusions

5.1 Summary

This thesis examines the application of minimum Hellinger distance classification to underwater acoustic signals. It begins with an explanation of the mathematics of the classifier and a comparison to a maximum likelihood method. The classifier is then applied to synthetic acoustic data. Finally, the classifier is applied to real acoustic data.

In Chapter 2 minimum distance classification is defined and its general properties are presented. Distances in statistics are then explored and the Minimum Hellinger Distance Classifier is motivated and developed as a particular type of minimum distance classifier. Some of the MHDC’s important properties are examined and a comparison of its performance to a classifier based on the log-likelihood ratio is presented. Contamination scenarios are presented and the MHDC is shown to have similar performance to the LLRC in the case of data with no contamination or inliers and superior performance in the case of outliers for Gaussian data sets. This is an expected result and it verifies the validity of the implementation of the classifier.

In Chapter 3 the MHDC’s application to synthetic acoustic data is examined and its performance compared to the LLRC. The affect of limited data on testing of the classifiers is examined and sub-sampling schemes are explored. It is found that interleaved sampling of the data improves results due to correlation in the signal. The affect of uncertainty in environment and source depth on statistical models is then examined. Several scenarios comprising variations on the amount of uncertainty in the statistical models are performed and the MHDC is shown to have superior performance to the LLRC on synthetic acoustic data when the acoustic situation is uncertain. The robustness properties of the MHDC explained in Chapter 2 are evident in the simulation results.

In Chapter 4 the classifier is evaluated when applied to received data from the SwellEx-96 experiment and an appropriate model of its acoustic environment. SwellEx-96 and the environmental model are explained and the challenges in processing this data set are explored. The
MHDC’s performance is then compared to that of a log-likelihood processor in a parallel of the
analysis in Chapter 3. The performance of the MHDC is seen to be equivalent to that of the
LLRC with the acoustic data that was tested. This result reinforces the equivalence of the MHDC
and LLRC when models are accurate. Further experiments with more data and more variation
in accuracy of environmental models are required to extrapolate these results.

5.2 Future Research

5.2.1 Metrics

A strength of the minimum distance method is that the metric used in classification can be
tailored to the problem. A different metric tailored to the mismatch inherent in the statistical
modeling process could improve the performance of the classifier. [7] illustrates several metrics
similar to the Hellinger distance that could better suit the strengths and weaknesses of the
statistical models used in the classification. A thorough understanding of the mismatches inherent
in the applied statistical modeling techniques is required to fully exploit the metric used in
classification. Therefore research in the statistical modeling process is vital to improving the
application of this type of minimum distance method.

5.2.2 Models

Improvements to the statistical models used to represent the received data and class models used
in testing could both be improved.

One of the simplest improvements to the statistical models could be made by examining the
density estimation methods used. Kernel Density Estimates are ideal for some situations, but the
choice of kernel and kernel bandwidth can greatly affect the shape of the estimate. In addition to
tweaking of those parameters, Kernel Density Estimates can require large amounts of storage and
processing, making them less than ideal for resource-scarce applications. Exploration of different
kernels and methods of choosing bandwidth could yield improved results. Also, a simpler model
such as Gaussian mixtures could possibly provide a much simpler and smaller statistical model.
In general a more compact model trades complexity in storage and processing for accuracy in
statistical description. This tradeoff should be explored.

A more complex yet possibly more fruitful area for improvement is the incorporation of
acoustic simulations into the statistical class models. In this work a very simple Monte Carlo
experiment is performed on two parameters. In a real situation many more parameters could
be involved. This becomes prohibitive for Monte Carlo methods due to the exponential growth
of required simulations. Several options are available to alleviate this problem. Alternative
statistical predictions of acoustic environments exist and could possibly be exploited. In addition,
studies of the variability in acoustic propagation due to uncertainty or variability in acoustic
environmental parameters can be studied and this knowledge used to decrease the number of
degrees of freedom required in Monte Carlo simulations.
Density Estimation

Density estimation is an important step in the Minimum Hellinger Distance Classifier and something to be examined. One representation of the probability density of the data is a histogram. Data is placed into bins and the number of points in each bin is plotted as the magnitude of that bin. This method is sensitive to the bin size and spacing and the amount of data, requiring adaptation of its parameters based on the data to form a reasonable density estimate. An alternative is a Gaussian mixture model. In this method a series of Gaussian distributions are fit to the data, each with its own mean and variance. This method is limited in that the number of Gaussian curves fit determines the number of modes in the data and it relies on the assumption that the modes will be mound-shaped like the Gaussian curves fit to them. Its advantage is that it takes a very small amount of storage as only the mean and variance of each Gaussian must be stored. A third alternative, kernel density estimates, gives more flexibility at the expense of storage and computational complexity.

A.1 Kernel Density Estimates

A kernel density estimate creates densities that are easy to work with and can be very accurate if the bandwidth and number of kernels are chosen well. They can however result in some inaccuracies. If the bandwidth is too large a narrow peak can be spread or a small peak could be completely missed. Too small of a bandwidth could result in a peaky distribution that does not have the nice smoothed character that is desired. Inaccuracies can also result from not having enough kernels used to estimate the data. Too few kernels could result in missing modes. In the extreme, a single kernel would simply fit a Gaussian distribution to the data, even if it were multi-modal. Further discussion of kernel density estimates can be found in [21].

The kernel density estimate $\hat{f}(x)$ uses a sum of normalized kernels on the data to create a smoothed estimate of the density of the data,
\[ \hat{f}(x) = \frac{1}{N h} \sum_{i=1}^{N} K \left( \frac{x - x_i}{h} \right) \] (A.1)

where \( N \) is the number of kernels in the estimate, \( h \) is the kernel’s bandwidth and \( K(x) \) is the kernel. Typically a Gaussian kernel is used with unity variance and zero mean

\[ K(x) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{x^2}{2} \right) \] (A.2)

The result is a smooth estimate of the probability density function.

With the Gaussian kernel chosen, two parameters must be determined to properly execute the kernel density estimate. First \( N \), the number of kernels in the estimate, is typically chosen to be the number of data points in the series for which a probability density function is being estimated, i.e. the length of \( x \). The second parameter, \( h \) is more difficult to choose and depends on the data. The bandwidth choice in this work is based on the optimal choice for a Normal distribution as described in [22]. With this method

\[ h = \left( \frac{4}{3n} \right)^{1/5} \sigma \] (A.3)

where \( \sigma \) is an estimate of the standard deviation of the distribution. While this choice is clearly sub-optimal for multi-modal distributions due to the assumption of Normality, it is a reasonable choice for the unimodal distributions seen in this thesis. To get a value for \( \sigma \) a robust estimator called the median absolute deviation estimator is used where

\[ \tilde{\sigma} = \frac{\text{median} \{ |y_i - \tilde{\mu}| \}}{0.6745} \] (A.4)

and \( \tilde{\mu} \) is the median of the sample.

This method produces a bandwidth that is slightly larger than necessary and therefore induces some smoothing on the estimates.

Fig. A.1 shows an example of summed Gaussian kernels creating a smooth probability density function. This is an example designed to show that the sum of several Gaussian kernels can produce a smooth probability density function and is not made with a properly calculated bandwidth from data. An example of a kernel density estimate applied to data can be found in Fig. A.2.

Fig. A.2 shows an example of a kernel estimate fit to a multi-modal density. This multi-modal density is a sum of six Gaussian densities, normalized to produce a proper probability density function. Each has separate means and variances: \( \mu_1 = 0, \sigma_1 = 3, \mu_2 = 15, \sigma_2 = 4, \mu_3 = 30, \sigma_3 = 3, \mu_4 = 45, \sigma_4 = 6, \mu_5 = 50, \sigma_5 = 4, \mu_6 = 75 \) and \( \sigma_6 = 2 \). A histogram is plotted in blue in addition to the true probability density in green and a kernel density estimate in red. Some inaccuracies due to smoothing can be seen in the estimate; however the character of the true probability density function is maintained.

Using the kernel density methods one can accurately estimate the probability density function
Fig. A.1. A weighted sum of kernels forms a probability density function in a kernel density estimate.

Fig. A.2. Histograms and kernel density estimates both approximate the probability density functions of random data.
of a time series without explicit knowledge of its underlying density.
Appendix B

Matlab Code

This appendix contains a collection of representative Matlab code. It is not an exhaustive collection of all code, but rather a selection that demonstrates the steps needed to set up the classifier. This code assumes a Monte Carlo simulation has been run, producing a collection of RAM results that have been imported into Matlab.

B.1 Forming models from TL fields

This code takes a collection of 441 environments and creates kernel density estimate-based probability density functions representing uncertainty in received amplitudes at certain depths. There are several helper functions at the end that load the data from a RAM run and generate various vectors needed for the kernel density estimates.

```matlab
% I form a PDF for each environment then average over all environments.
clear all;close all;clc;

% depths for single depth models
z_deep_sing=70;
z_shal_sing=30;

% depths for depth window models
z_deep-wide=[65 75];
z_shal-wide=[25 35];

% generate the domain of the models
nbins=500;
x_model=models_gen_xmodel(z_deep_sing, z_shal_sing, nbins);

% Generate Monte Carlo environment models %
```
% number of environments in the monte carlo run
numEnvs=21*21;
singleEnv=221;

% initialize variables
p_deep_sing=zeros(1,nbins);
p_shal_sing=zeros(1,nbins);
p_deep_wide=zeros(1,nbins);
p_shal_wide=zeros(1,nbins);

mybar=waitbar(0,'Making Monte Carlo Models');
for ii=1:numEnvs
    waitbar((ii-1)/numEnvs,mybar);
    set(mybar,'name',
        ['Environment ' num2str(ii)]);
    eval(['load RAMResults\GENLMIS_MC' num2str(ii) '.mat;']);
    % create models for a single depth
    [p_deep p_shal]=models_gen_pdf(TL, z, z_deep_sing, z_shal_sing, x_model);
    if ii==singleEnv %grab the single environment model
        save models\models_simple_sing.mat x_model p_deep p_shal;
    end
    % add it to the rest of the models, averaging
    p_deep_sing=p_deep_sing+p_deep/numEnvs;
p_shal_sing=p_shal_sing+p_shal/numEnvs;
    % create models for a depth window
    [p_deep p_shal]=models_gen_pdf(TL, z, z_deep_wide, z_shal_wide, x_model);
    if ii==singleEnv %grab the single environment model
        save models\models_simple_wide.mat x_model p_deep p_shal;
    end
    % add it to the rest of the models, averaging
    p_deep_wide=p_deep_wide+p_deep/numEnvs;
p_shal_wide=p_shal_wide+p_shal/numEnvs;
    clear TL r dr rmax z dz zmax freq;
end
close(mybar);

p_deep=p_deep_sing;
p_shal=p_shal_sing;
save models\models_MC_sing.mat x_model p_deep p_shal;

p_deep=p_deep_wide;
p_shal=p_shal_wide;
save models\models_MC_wide.mat x_model p_deep p_shal;

function x_model=models_gen_xmodel(zdeep, zshal, nbins)
% generates the domain of the statistical models from
% the range of the received data
[deep shal] = received_load(zdeep, zshal);
fprintf('Determining x_model...
');
x_model=linspace(min([min(deep(:)) min(shal(:))]),
    max([max(deep(:)) max(shal(:))]),nbins);
end

function [p_deep p_shal]=models_gen_pdf(TL, z, zdeep, zshal, x_model)
% function [p_deep p_shal]=models_gen_pdf(TL, z, zdeep, zshal, x_model)
% generates a deep and shallow model from the TL data

[deep_data shal_data] = RAM_load(TL, z, zdeep, zshal);

% do a kernel density estimate
p_deep=ksdensity(deep_data(:),x_model);
p_shal=ksdensity(shal_data(:),x_model);
end

function [deep_data shal_data] = RAM_load(TL, z, zdeep, zshal)
% function [deep_data shal_data] = RAM_load(TL, z, zdeep, zshal)
% inputs:
% TL TL field from which data will be extracted
% z true depths
% zshal depth(s) for shallow data
% zdeep depth(s) for deep data
% outputs:
% deep_data TL data at depths zdeep
% shal_data TL data at depths zshal

%convert depths to indices
if length(zdeep)>1 || length(zshal)>1
    deep_indices=find(z==zdeep(1)):find(z==zdeep(2));
    shal_indices=find(z==zshal(1)):find(z==zshal(2));
else
    deep_indices=find(z==zdeep);
    shal_indices=find(z==zshal);
end

% extract TL data
deep_data=TL(deep_indices,:);
shal_data=TL(shal_indices,:);

% normalize by MSE...
deepl_data = deep_data/sqrt(var(deep_data(:))+mean(deep_data(:))^2);
shal_data = shal_data/sqrt(var(shal_data(:))+mean(shal_data(:))^2);
end
B.2 Running the classifiers

This is the heart of the classification code. Models and data are loaded and processed in \( N \) sections. Both the LLRC and MHDC are implemented. This code can do several different styles of data sampling and implements some of the tests described in Chapter 3. Subfunctions for loading models and data and saving results are omitted. The product of this code is a set of detection statistics (or scores for the MHDC) that may be used to create ROC curves.

```matlab
% This will process all received data, using a minimum
% hellinger distance classifier or a log-likelihood based classifier. The data
% from the RAM runs is compared to the deep and shallow prototypes.

clear all; close all; clc;

doLR=1;
doMH=1;

%% SETTING THINGS UP
Narray=[100 200 300 400 500];
zdeep=70;
zshal=30;

for rand_sample_type=0:4
    mybar=waitbar(0, 'Processing Syn vs. Syn');
    for ii=0:3
        % make knownDepth and knownEnv go through the 4 possible combinations of 0
        % and 1
        knownDepth=~bitshift(bitand(2,ii),-1);
        knownEnv=~bitand(1,ii);

        % detect arrays should have 1 column per # of sections and each column should
        % have enough elements to hold N detection statistics at each frequency.
        % we make it big enough to hold the max N cause we can't have each row a
        % different size.
        det_deep_LR=cell(length(Narray),1);
det_shal_LR=cell(length(Narray),1);
det_deep_MH=cell(length(Narray),1);
det_shal_MH=cell(length(Narray),1);

        [p_deep p_shal x_model] = models_load(knownEnv,knownDepth);
        [d_deep d_shal] = received_load(zdeep,zshal);

        for Nindex=1:length(Narray)
            N=Narray(Nindex);
            set(mybar,'name',['N=' num2str(N)]);
            waitbar((ii/4)+(Nindex-1)/(4*length(Narray)),mybar);
            if doMH
```
function [score_deep score_shal] = process_N_sections(d_deep, d_shal, p_deep, p_shal, x_model, N, doLR, randSampleType)
% function [score_deep score_shal] = ...
% process_N_sections(d_deep, d_shal, p_deep, p_shal, x_model, ... 
% N, doLR, randSampleType)
% % calculates the scores for the loaded data for N pieces
% % inputs:
% % d_deep received data from deep
% % d_shal received data from shallow
% % p_deep model data from deep
% % p_shal model data from shallow
% % x_model domain of model pdfs
% % N number of sections to divide experiment data into
% % doLR boolean value: true for LR, false for MHDC
% % doRandSample Sets the type of sampling to do
% % outputs:
% % score_deep score for each section in the deep data
% % score_shal score for each section in the shallow data

plotStuff=0;

% go ahead and calculate the likelihood ratio early on
if doLR
    L=log(p_deep./(p_shal+eps));
end
L(L>500)=500;
L(L<−500)=−500;
end

if doLR
fprintf('Processing LLRC in %d sections...
',N);
else
fprintf('Processing MHDC in %d sections...
',N);
end

% total number of points that can be drawn from
total_data_length=length(d_shal);
% number of points in each section
section_length=floor(length(d_shal)/N);
% then we have to form N PDF's from the experiment and calculate
% the MHDC scores for the deep and shallow portions
score_deep=zeros(1,N); % initialize score variable
score_shal=zeros(1,N); % initialize score variable
for ii=1:N
    if randSampleType==0 % use all data
        indices=1:length(d_deep);
    elseif randSampleType==1 % interleaved sampling (length/N)
        indices=(1+section_length*(ii-1)):section_length*ii;
    elseif randSampleType==2 % random sampling (length/N)
        indices=floor(total_data_length*rand(1,floor(total_data_length/N)))+1;
    elseif randSampleType==3 % random sampling (length)
        indices=floor(total_data_length*rand(1,total_data_length))+1;
    elseif randSampleType==4 % offset interleaved sampling
        % i-section # section=(i,N+1,2N,i,etc.)
        indices=ii:N:total_data_length;
    else
        error('Need to choose randSampleType between 0 and 4');
    end

d_deep_section=d_deep(indices);
d_shal_section=d_shal(indices);
if plotStuff
    figure(17);hold on;
    plot(x_model,ksdensity(d_deep_section,x_model),'b');
else
    if doLR
        % compute likelihood ratio (summing over entire section)
        score_deep(ii)=sum(interp1(x_model,L,d_deep_section));
        score_shal(ii)=sum(interp1(x_model,L,d_shal_section));
    else
        % form the kernel density based on the section
        p_deep_rec=ksdensity(d_deep_section,x_model);
        p_shal_rec=ksdensity(d_shal_section,x_model);

        % calculate score for the section
        score_deep(ii)=helldist(x_model,p_deep_rec,p_shal) - ...
helldist(x,model,p_deep_rec,p_deep);
score_shal(ii)=helldist(x,model,p_shal_rec,p_shal) - ...
helldist(x,model,p_shal_rec,p_deep);
end
end
end
end

function H = helldist(X,P,Q)
% it is assumed that P and Q are PDFs sampled at the points in X
    dX=max(diff(X));
    H=norm(sqrt(P*dX)-sqrt(Q*dX))/sqrt(2);
end

B.3 Calculating \( p_{fa} \) and \( p_d \)

This section of code converts detection statistics (for the LLRC) or scores (for the MHDC) into \( p_{fa} \) and \( p_d \) for creating ROC curves. It is assumed that \( H_1 \) is the correct hypothesis. \( L_1 \) and \( L_2 \) contain the detection statistics or scores for a realization from \( H_1 \) and \( H_2 \), respectively, run through the classifier.

function [Pd Pfa] = det_stat_to_ROC(L1, L2)
% function [Pd Pfa] = det_stat_to_ROC(L1, L2)
% Generates [Pd,Pfa] for a ROC curve from a pair of detection statistic
% vectors L1 and L2 corresponding to hypotheses H1 and H2,
% respectively.
% H1 is assumed to be the correct hypothesis.
% if nargin<2
%     error('Need both L1 and L2.');
% end
    N=length(L1);
    if N ̸= length(L2)
        error('L1 and L2 must be the same length.');
    end

    L1_sort = sort(L1(:)); % First sort the data
    L2_sort = sort(L2(:)); % and orient the vectors correctly

    % Then generate the ROC curves. There will be as many points in
    % the ROC curve as there are detection statistics.

    Pfa = linspace(0,1,N); % Pfa is a linear from 0 to 1 with N points
    Pd = zeros(size(Pfa)); % initialize Pd
    for L = 1:N; % Step through L2. Pd is the fraction of points
% in L1 that are greater than the current point.
Pd(this_L) = sum(L1_sort > L2_sort(this_L))/N;
end
Pd = sort(Pd);
end


