

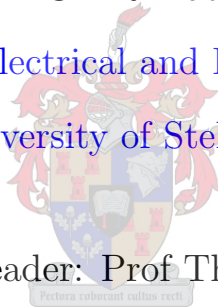
Lung Health Diagnosis Through Cough Sound Analysis

GHR Botha

Department of Electrical and Electronic Engineering

University of Stellenbosch

Study leader: Prof Thomas Niesler



Thesis presented in partial fulfillment of the requirements for the degree
of Master of Science in Engineering at Stellenbosch University

M.Eng E&E

March 2017

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

.....

Signature

.....

Date

Acknowledgements

I would like to thank my mother Therese, and my father Louis, for their continuous support and encouragement.

I would also like to thank Telkom and the Centre of Excellence for financial support during my postgraduate studies.

Abstract

This study investigates a simple and easily applied tool for TB screening based on the analysis of cough audio and objective clinical measurements.

Tuberculosis is one of the most lethal diseases worldwide. There are various diagnosis methods for TB. However, in lower income areas, clinics lack funds to afford expensive equipment and employ the trained experts needed to interpret results.

A database of cough audio recordings and clinical measurements was collected for this study. An automatic annotation system was developed using hidden Markov models (HMMs). The frame-accuracy of the annotation system is 87.16%.

For audio based classification we considered logistic regression and Gaussian mixture models (GMMs). We found that filterbank energy features outperformed MFCC features when used for audio classification, which could indicate that cough audio contains information relevant to TB diagnosis that is not perceivable by the human auditory system. Feature selection was used to investigate the importance of different frequency bands for classification and, it was found that the optimal results were achieved when combining features from the human vowel range (below 1000Hz) with features from high frequency ranges.

As the main metric of evaluation, we used the area under the receiver operator characteristic curve (AUC). This metric was chosen because it is not affected by class imbalance in the dataset. Our best reported AUC was 94.94%, with a standard deviation of 4.62%, which was obtained using a set of just 5 filterbank energies. We also showed that audio based classification obtains a higher AUC than classification on objective clinical measurements (meta data).

Finally, we found that combining the audio and meta data classifier results using classifier fusion improved how well the model generalizes. By combining the best audio classifier with the best meta data classifier, we obtained a sensitivity, specificity, accuracy, AUC and kappa of 82.35%, 80.95%, 81.58%, 94.34% and 0.6867 respectively.

Opsomming

Hierdie studie ondersoek 'n eenvoudige en makliktoegepaste instrument vir die skandering van tuberkulose (TB), gebaseer op die analise van hoes-audio en objektiewe kliniese metings. Tuberkulose is wereldwyd een van die dodelikste siektes. Daar is verskeie metodes vir die diagnose van TB. In laer-inkomste areas is daar egter gebrekkige befondsing vir duur toerusting en die aanstelling van opgeleide kundiges om toetsuitslae te interpreteer.

'n Databasis van hoes-audio opnames en kliniese metings is vir hierdie studie versamel. 'n Outomatiese annotasiesistelsel is ontwikkel deur versteekte Markov modelle (HMMs) te gebruik. Die beramingsakkuraatheid vir die annotasiesistelsel is 87.16%.

Vir audio-gebaseerde klassifikasie het ons logistiese regressie en Gaussiese vermengingsmodelle (GMMs) gebruik. Ons het gevind dat filterbank energie kenmerke meer doeltreffend as MFCC kenmerke is wanneer dit vir audio-klassifikasie gebruik is, wat kan aandui dat hoes-audio inligting relevant tot TB diagnose bevat wat nie deur die menslike gehoorstelsel geregistreer kan word nie. Funksie seleksie is gebruik om die belangrikheid van verskillende frekwensiebande vir klassifikasie te ondersoek en daar is gevind dat die optimale uitslae bereik is wanneer funksies van die menslike vokaalreeks (onder 1000Hz) met funksies van ho frekwensiereekse gekombineer is.

Ons het die area onder die ontvangers operator eienskap kurwe (ROC AUC) as die hoofmatriks van evaluering gebruik. Hierdie matriks is gekies omdat dit nie deur klaswanbalans in die datastel geaffekteer word nie. Ons mees doeltreffende AUC was 94.94%, met 'n standaardafwyking van 4.62%, wat verkry is deur 'n stel van slegs 5 filterbankenergie te gebruik. Ons het ook gewys dat audio-gebaseerde klassifikasie 'n hoër AUC bereik as klassifikasie op objektiewe kliniese metings (metadata).

Laastens het ons gevind dat die kombinerings van die audio en metadata klassifiseringsuitslae deur klassifiseringsfusie die veralgemening van die model verbeter het. Deur die beste audio klassifiseerder met die beste metadata klassifiseerder te kombineer het ons 'n sensitiwiteit, spesifisiteit, akkuraatheid, AUC en kappa van 82.35%, 80.95%, 81.58%, 94.34% en 0.6867 onderskeidelik verkry.

Contents

1	Introduction	15
1.1	Problem Statement	15
1.1.1	Objective	15
1.1.2	Data	15
1.1.3	Scope	16
1.2	The Respiratory System	16
1.3	Causes of Coughing	17
1.3.1	Diagnosis of coughs	17
1.4	The Human Auditory System	18
1.4.1	Anatomy of the Ear	18
1.4.2	The Range of Human Hearing	20
1.5	Tuberculosis	21
1.5.1	Tuberculosis in South Africa	21
1.5.2	Pathology of Tuberculosis	21
1.5.3	Symptoms of Tuberculosis	22
1.5.4	Diagnosis of Tuberculosis	22
1.5.5	Treatment of Tuberculosis	24
1.6	Summary	24
2	Literature Review	25
2.1	Cough Detection	25
2.1.1	Specific Studies	25
2.2	Cough Classification	27
2.2.1	Data Acquisition	28
2.2.2	Feature Extraction & Selection	28
2.2.3	Specific Studies	29
2.3	Summary	33

3	Classification and Evaluation Methods	34
3.1	Hidden Markov Models (HMMs)	34
3.1.1	Markov Models	34
3.1.2	Hidden Markov Models	35
3.2	Logistic Regression	42
3.3	Gaussian Mixture Models	47
3.4	Evaluation Methods	50
3.4.1	Metrics	50
3.4.2	Cohen’s Kappa Coefficient	52
3.4.3	Receiver Operator Characteristic Curve Analysis	53
3.4.4	Cross Validation	55
3.5	Summary	57
4	Data Acquisition	58
4.1	Data Acquisition	58
4.1.1	Recording Setup	59
4.1.2	Audio Dataset	60
4.1.3	Clinical Dataset	61
4.2	Database Summaries	63
4.3	Summary	64
5	Data Preparation	65
5.1	Data Pre-processing	65
5.1.1	Audio Cleaning	65
5.1.2	Normalization	66
5.2	Annotation and Segmentation	68
5.2.1	Manual Annotation	68
5.2.2	Automatic Segmentation	69
5.3	Summary	74
6	Cough Classifier Design and Evaluation	75
6.1	System Overview	75
6.2	Feature Extraction	75
6.2.1	Log Filterbanks	76
6.2.2	Mel-frequency Cepstral Coefficients (MFCCs)	77
6.2.3	Zero Crossing Rate (ZCR)	81
6.2.4	Kurtosis	81
6.3	Experimental Setup	82
6.3.1	Constructing Feature Datasets	82
6.4	Experimental Evaluation	83

6.4.1	Classification Performance	87
6.4.2	Meta Data Classifier	91
6.4.3	Classifier Fusion	92
6.5	Discussion and Further Investigation	94
6.6	Summary	100
7	Conclusion and Future Development	103
7.1	Future Development	104
A	Audacity Audio Editing Tool	106
	References	106

List of Figures

1.1	The human respiratory system. Modified from [3].	17
1.2	Anatomy of the ear, taken from [10].	19
1.3	A loudness curve, taken from [14].	20
2.1	Example of a cough waveform with three phases indicated.	29
2.2	Recording setup used for pneumonia cough classification in [38].	31
3.1	A three state Markov model.	35
3.2	Hidden Markov model with 3 states $\{s_1, s_2, s_3\}$ and three observable values $\{x_1, x_2, x_3\}$	36
3.3	Logistic sigmoid function	42
3.4	L2 regularization visualization modified from [45].	46
3.5	Single Gaussian distribution	47
3.6	Example of multi modal Gaussian distributions.	48
3.7	Hypothetical ROC curve [50].	54
3.8	Graphical representation of K-fold validation.	55
3.9	Parameter selection flow diagram.	56
4.1	Recording setup at the Brooklyn Chest Health Clinic.	59
5.1	Waveform before and after manual trimming. The red segments indicate sections that will be removed from the top waveform, resulting in the bottom waveform.	66
5.2	Computation of mel frequency cepstral coefficients (MFCCs).	70
5.3	HMM State Diagram	71
6.1	Cough classifier evaluation flow diagram.	75
6.2	Linearly-spaced triangular filterbank.	76
6.3	Feature extraction by log filterbanks. The area under the curve in the bottom right graph constitutes a single feature.	77
6.4	MFCC calculation flow diagram.	78
6.5	Mel-scaled filters	79
6.6	An overview of the classification and evaluation process.	85

LIST OF FIGURES

6.7	Mean ROC curve of the best performing logistic regression model in terms of ADS AUC with the TIS based curve included.	89
6.8	Mean ROC curve of the best performing logistic regression model in terms of TIS with the ADS based results included.	89
6.9	Mean ROC curve of best performing GMM model in terms of ADS and TIS evaluation. The best ADS and TIS results are obtained from the same feature extraction parameters.	91
6.10	Individual frequency band ADS AUC results for F:140.	95
6.11	Vowel and consonant fundamental frequency ranges (reproduced from [60]).	95
6.12	Classification performance on filterbank segments.	97
6.13	Individual frequency band ADS AUC results for F:140 with bootstrap means and standard deviations.	98
6.14	Performance in terms of ADS AUC for greedy search feature selection. . .	99
6.15	Forward selection algorithm results.	100
A.1	Audacity GUI showing an already loaded recording.	107
A.2	Example of output text file containing annotations.	109

List of Tables

1.1	Comparison of Current TB Diagnosis Methods.	22
2.1	Classification Results presented in [33].	26
3.1	An example of a confusion matrix.	51
3.2	Spam Filter Confusion Matrix	52
4.1	Complete Audio Dataset	63
4.2	Cough Classifier Dataset \mathcal{D}_{CC}	63
4.3	Automatic Annotator Dataset \mathcal{D}_{AA}	63
4.4	Clinical Data Information	63
5.1	Segmented \mathcal{D}_{CC} Corpus	69
5.2	Segmented \mathcal{D}_{AA} Corpus	69
5.4	HMM Frame-based Accuracies: Validation set results	73
5.5	HMM Frame-based Accuracies: Test set results	73
5.3	HMM hyper-parameter scope	73
6.1	Feature extraction hyper parameters.	82
6.2	Values considered for hyper-parameter optimization using grid search.	84
6.3	Logistic regression patient-level results. Refer to Table 6.1	88
6.4	Gaussian mixture model patient-level results. Refer to Table 6.1 for clarification of the abbreviations used.	90
6.5	Meta-classifiers results	92
6.6	Results of classifier fusion.	93
6.7	Classifier combination methods results for model trained on 5 best features.	101

List of Algorithms

1	Forward Algorithm	39
2	Viterbi Algorithm	40
3	Backward Algorithm	41
4	Baum-Welch Algorithm	42

Nomenclature

Acronyms

ADS	Average Diagnosis Score
ANN	Artificial Neural Network
AUC	Area Under the Curve
BGS	Bispectrum Score
BMI	Body Mass Index
CT	Computed Tomography
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DT	Decision Trees
FEV	Forced Expiratory Volume
FF	Formant Frequencies
FFT	Fast Fourier Transform
FPR	False Positive Rate
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
IID	Independent and Identically Distributed
LCM	Leicester Cough Monitor
LOOV	Leave One Out Validation
LRM	Logistic Regression Classifier

LSTM	Long Short-term Memory
MDR-TB	Multi-Drug Resistant TB
MFCC	Mel Frequency Cepstral Coefficients
MLP	Multilayer Perceptrons
MTB	Mycobacterium Tuberculosis
MUAC	Mid Upper Arm Circumference
NGS	Non-Gaussianity Score
PCI	Pneumonic Cough Index
PCR	Polymerase Chain Reaction
ROC	Receiver Operator Characteristic
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
TB	Tuberculosis
TIS	Tuberculosis Index Score
TPR	True Positive Rate
UCT	University of Cape Town
ZCR	Zero Crossing Rate

Chapter 1

Introduction

1.1 Problem Statement

There are various diagnosis methods for tuberculosis (TB). In lower income areas, clinics lack funds to afford expensive equipment and employ the trained experts needed to interpret results. Therefore clinics resort to cheaper methods of diagnosis, such as sputum culture tests which can take up to 6 weeks to be conclusive [1]. The longer diagnosis process causes individuals who are sick and highly contagious to infect more people. This is one of the reasons why TB, a curable disease, is still a major issue worldwide.

TB usually manifests itself in the upper part of the lungs and distinctly forms small cavities as the bacteria deteriorates the lung tissue. Thus, the coughing sound of an individual with pulmonary TB could contain information indicative of the infection. This information could be used to develop a tool to aid in the diagnosis process that is affordable, would provide quick results and does not require expert training to operate.

1.1.1 Objective

The aim of this project is to investigate the possibility of developing a system to detect cough audio patterns associated with lung diseases, such as TB, which can be used to "rule-in" individuals with an abnormal cough for diagnostic testing.

1.1.2 Data

A database of cough sounds will be compiled for subsequent statistical analysis and machine learning. Previous studies indicate that we should aim to collect data from 100 patients, however recent studies have shown success with as few as 18 patients [2]. From each patient, we aim for 10 coughs to be recorded. Data will be collected from patients with pulmonary TB and from healthy individuals. If possible, the dataset will be expanded

1.2 The Respiratory System

to include the recruitment of patients with other lung diseases (such as bronchitis).

1.1.3 Scope

The project will focus on the classification of coughs as either TB or non TB related through cough audio analysis. As a baseline, the system should be able to distinguish between coughs of healthy patients and coughs of patients with TB. This includes the design of a system to automatically detect cough events from continuous audio in a closed environment. If possible, the scope will be expanded to include the analysis of clinical measurements, which could be combined with audio based classification techniques to achieve higher classification accuracies.

1.2 The Respiratory System

The process of catering for the human body's need for oxygen and the management of discharged carbon dioxide is referred to as respiration [3]. Oxygen is a vital resource for the body, because all cells in the human body require oxygen to function properly.

The respiration process begins by inhalation through the mouth or nostrils. Air is then transferred through the larynx and trachea (**Figure 1.1(a)**) into two smaller tubes called bronchi, which develop into the bronchial tubes or "bronchioles". The bronchioles, seen in **Figure 1.1(b)**, feed the air into the lungs and are connected to small sacs called alveoli (**Figure 1.1(c)**), where the oxygen diffuses into the red blood cells in the bloodstream [4].

1.3 Causes of Coughing

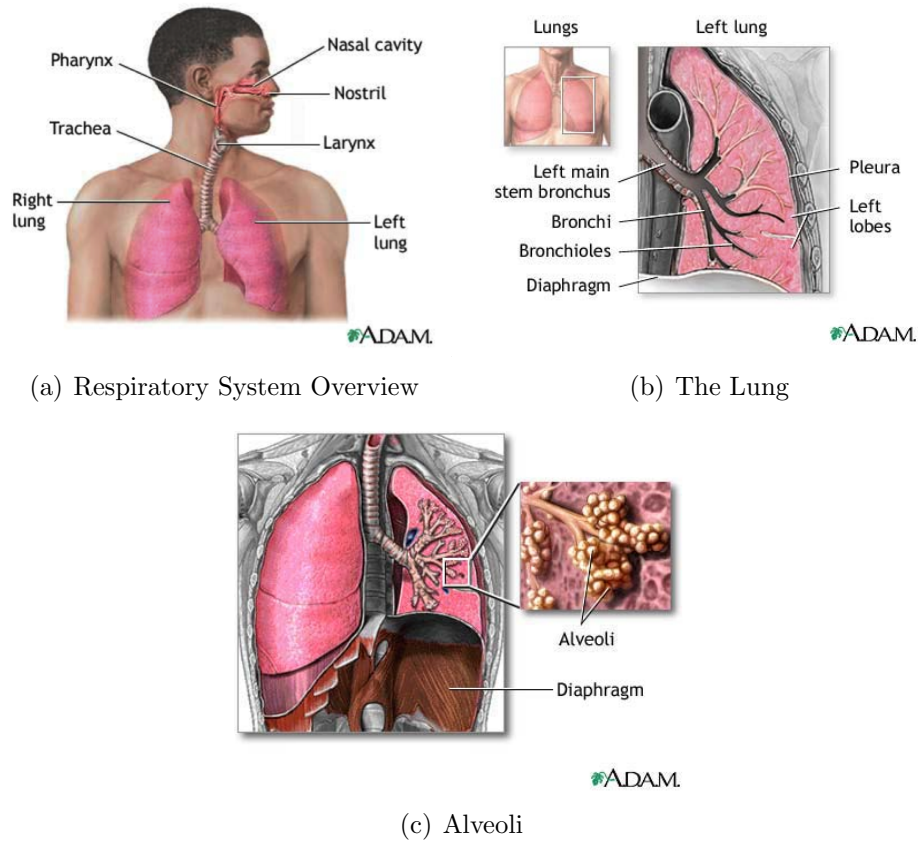


Figure 1.1: The human respiratory system. Modified from [3].

Blood is circulated through the body by the heart, delivering oxygen to the organs. After oxygen has been delivered, carbon dioxide is absorbed from the organs back into the blood. The blood cells carrying carbon dioxide return to the lungs where the carbon dioxide is expelled from the body through exhalation [5].

1.3 Causes of Coughing

Coughing is the body's natural expulsive reflex to the presence of a physical or chemical substance that is foreign to the respiratory system. In anatomical terms, a cough is caused when cough receptors are stimulated by foreign substances. The cough receptors are located in various parts of the respiratory system (nose, trachea, diaphragm etc.) and are connected to a portion of the medulla oblongata called the 'cough center'. When cough receptors are stimulated, the cough center sends impulses to the respiratory system to contract in such a way as to expel the foreign substance [6].

1.3.1 Diagnosis of coughs

Currently the norm when diagnosing a cough is for the doctor to ask the patient to describe the cough (e.g. dry/wet, severity, duration etc.). In addition, the doctor will listen

1.4 The Human Auditory System

to the patient cough and deduce his/her own description. This method relies heavily on the ability of the patient to correctly describe the cough and on the experience and hearing ability of the doctor.

Information about the wetness/dryness of a cough could narrow down the possible illnesses causing the cough. A wet cough would typically be associated with illnesses manifesting in the lower part of the respiratory tract, such as bronchitis, pneumonia and asthma. For a dry cough the absence of mucus is a descriptive feature and is usually associated with allergic diseases and sinusitis [7]. Tuberculosis patients generally experience a wet cough.

Recently, researchers have attempted to automatically classify coughs as wet or dry [7] [8]. Patients were recorded using bed-side microphones. These recordings were subsequently analysed to train an automatic classifier. However, limited research has been done to automatically diagnose specific illnesses through cough sound analysis. One such study is discussed in **Section 2.2**, where a system is designed to automatically diagnose pneumonia in children.

1.4 The Human Auditory System

A brief description of the human auditory system is provided in this section.

1.4.1 Anatomy of the Ear

The ear can be divided into three major parts: the outer ear, the middle ear and the inner ear, each part performing specific tasks.

The outer ear, seen in **Figure 1.2**, is responsible for channelling sound waves to the middle ear, while protecting the middle and inner ear. Additionally, the outer ear aids in sound localization [9]. The outer ear consists of the ear shell, also referred to as the pinna (not shown in the figure) and the ear canal.

1.4 The Human Auditory System

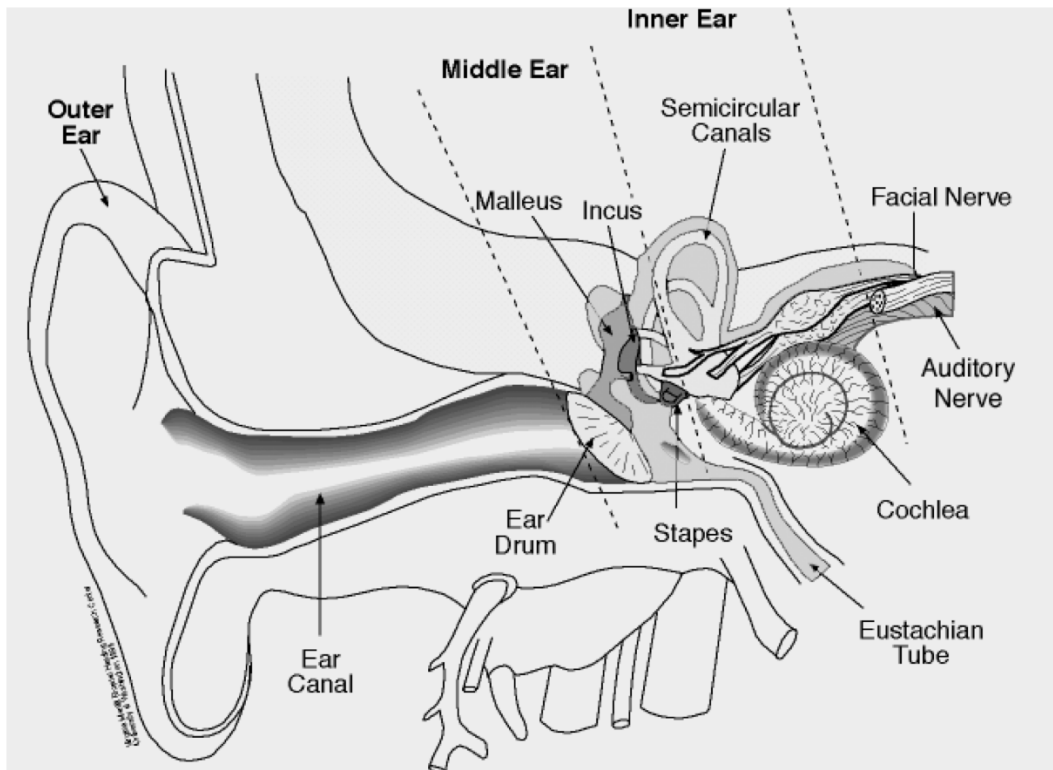


Figure 1.2: Anatomy of the ear, taken from [10].

The middle ear consists of the ear drum and the bone structure that keeps it in place (malleus, incus and stapes). It's main function is to perform impedance matching between the medium through which the sound waves are propagating (mostly air) and the fluid inside the cochlea. Without the impedance matching process, it is estimated that 99.9% of the sound energy would be lost [10].

Connected to the stapes of the middle ear is the cochlea, which constitutes the entire inner ear. The cochlea is a snail-shaped cavern, filled with approximately 30,000 hair cells that convert vibration into neural signals, transmitted through the auditory nerve. When rolled out, the hairs closer to the connection of the middle ear are short and stiff and become longer and softer the further away from the middle ear. This causes different hairs to have different resonant frequencies, which enables us to distinguish between different pitches. The spacing of these hairs follow a log-like distribution with hairs resonating at higher frequency more densely spaced and hairs that resonate at lower frequencies more sparsely spaced. This log-like spacing cause our hearing spectrum to be more sensitive at lower frequencies[11], [12].

The function of the cochlea may be considered to be similar to the Fourier transform, converting raw vibrational sound waves into neural signals in the frequency domain [12].

1.4.2 The Range of Human Hearing

Our hearing range is defined by the range of sound frequencies, measured in Hertz (Hz) that we can perceive without experiencing discomfort. The amplitude (or loudness) of sound is measured in decibels (dB). The maximum perceivable frequency range is approximately 20Hz to 20kHz for most humans. However, perceived sound is not only affected by the pitch (frequency), but also by the loudness (amplitude) [13].

Figure 1.3 shows a loudness curve, depicting the perceived loudness of sounds over the frequency range 20Hz to 20kHz. At 20kHz, the necessary loudness for perceived hearing approaches the threshold of pain, resulting in the end of the human audible spectrum [14]. Notice that within the range 3kHz to 5kHz, the threshold of hearing is at the lowest point, which means that our ears are most sensitive to sounds within this range. Human speech ranges between approximately 300Hz to 5kHz, including vowel and consonant sounds, which coincides with the most sensitive range of the human auditory system [15].

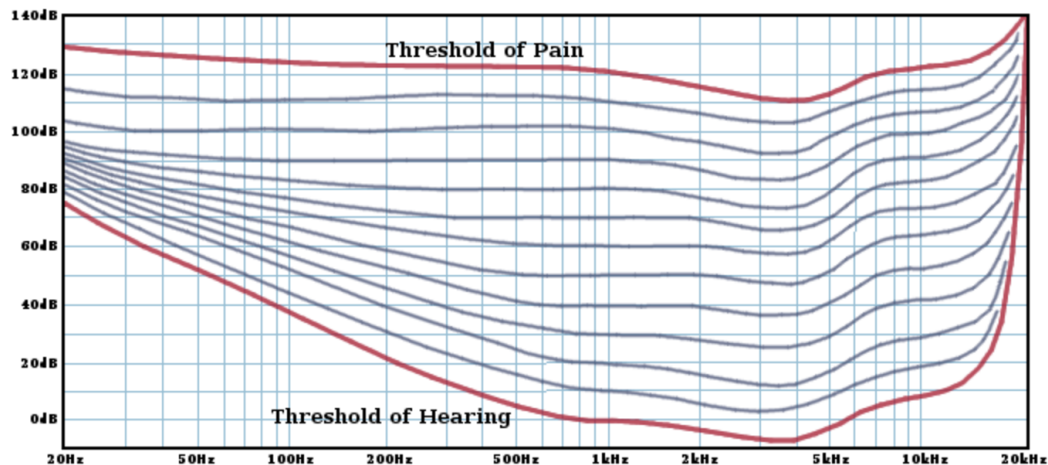


Figure 1.3: A loudness curve, taken from [14].

In order to interpret complex sounds we need to be able to distinguish between the various pitches that the sound is made of. For instance, if we want to enjoy listening to music, we need to be able to resolve the chords that are being played, or when recognizing speech, we are identifying the different vowels and consonants on a spectral level by distinguishing their frequency distribution.

When sound waves cause the hairs inside the cochlea to vibrate, the hairs that resonate with the vibrational frequency are stimulated, but some adjacent hairs are also stimulated in the process. This area that is stimulated is called the critical band. Two sounds that are within the same critical band cannot be distinguished in terms of pitch [16]. As mentioned, the spacing of the hairs in the cochlea is denser for higher- than for lower frequency resonating hairs. Therefore, lower frequency sounds have smaller critical bands and thus we are better equipped to distinguish pitch at lower frequencies than higher frequencies.

1.5 Tuberculosis

Alongside the Human Immunodeficiency Virus (HIV), TB is the most lethal disease worldwide. In 2014, 9.6 million people contracted the disease and 1.5 million people died as a result. India, Indonesia and China have the largest number of cases, with South East Asia and the Western Pacific regions accounting for 58% of the 9.6 million cases. However, Africa has the highest burden relative to its population [17].

It is estimated that one third of the world population carries a latent form of TB, which means that they have been infected by the bacterium, but have not yet fallen ill and cannot transmit the disease. A person infected with this latent form of TB has a 10% risk of falling ill within his or her lifetime. However, HIV, diabetes, malnutrition, smoking, and other diseases or activities compromising the immune system can cause this risk to increase dramatically [18].

1.5.1 Tuberculosis in South Africa

South Africa has the highest HIV positive TB incidence rate in the world, after larger countries like India and China [19] and the 6th highest incidence rate. In this context, incidence refers to the number of new cases reported in a country within a year. TB is especially lethal in the mining sector of South Africa. An estimated 80% of the South African population have latent TB, with the highest percentage found in the age group of 30-39 years, living in townships [20]. According to [21], 73% of TB positive patients are also HIV positive.

In the National Strategic Plan on HIV, sexually transmitted infections (STIs) and TB released by the South African National Aids Council (SANAC), one of four strategic objectives is to "Prevent new HIV, STI and TB infections" [22]. In this objective, it is stated that SANAC will aim to "maximize opportunities for testing and screening to ensure that everyone in South Africa is...screened for TB, at least annually...". Furthermore, the projected budget for this objective for 2016/2017 is listed as R20,946,090,000 (approximately R21 billion) and a total budget over all 4 objectives estimated at R32,247,500,000 ¹.

1.5.2 Pathology of Tuberculosis

Tuberculosis is a contagious and potentially deadly disease caused by infection with a bacterium called *Mycobacterium Tuberculosis* (MTB). TB is most often transmitted by inhaling tiny droplets containing MTB. These droplets are expelled by a person already infected with TB through coughing, sneezing or talking. A waxy coat allows the MTB to

¹Actual figures quoted in [22] are R20 946,90 billion and R32 247,5 billion respectively. However, we assumed that these amounts were not correctly reported and that the intended order of magnitude should have been millions. These assumptions could not be confirmed however.

survive in air for long enough to make transmission possible [23].

TB most commonly affects the lungs and is then referred to as *pulmonary tuberculosis*. However, once in the lungs TB, can affect any other organ of the body, then referred to as *extra-pulmonary* TB.

1.5.3 Symptoms of Tuberculosis

A characteristic symptom of TB is a chronic cough that worsens over a period of approximately 3 weeks. The worsening cough is linked to the MTB, which manifests primarily in the upper part of the lungs where more oxygen is present [24]. Over time, the MTB build up and form small nodules which irritates the lungs, causing a constant cough. Sometimes these nodules can erupt, leading to other complications such as internal infections and coughing up of blood or blood smeared sputum. Some other symptoms of TB include chest pains, unexpected weight loss, fever, night sweats and loss of appetite.

1.5.4 Diagnosis of Tuberculosis

There are several methods for diagnosing TB. The method used depends on the medical condition of the patient and the means of the institution where diagnosis takes place.

Table 1.1 shows a comparison of various methods for diagnosing TB in terms of time, cost, ability to detect active MTB and ability to determine whether a patient has drug resistant TB (drug susceptibility) [1].

Table 1.1: Comparison of Current TB Diagnosis Methods.

Test Name	Time frame	Cost	Active TB test	Drug susceptibility
Smear microscopy	< 24 hours	Low	No	No
Sputum culture	4 weeks	Medium	Yes	Yes
IGRAs	< 24 hours	High	No	No
Tuberculin skin test	48-72 hours	Low	No	No
Chest radiography	minutes	Very High	Yes	No
GeneXpert	<2 hours	High	Yes	Yes

- **Smear microscopy:** Sputum samples are collected, usually by taking swabs, and are viewed under a microscope to check for MTB. The process requires at least three sputum samples to provide an accurate diagnosis. This is a cheap and effective method for most cases, but it is not very accurate, with reported accuracies of 50-60% in well equipped laboratories [25].
- **Sputum culturing** This process involves growing the tuberculosis bacteria on a solid media from a sputum sample. When culturing TB samples, the sample needs

to be decontaminated to avoid overgrowth by other micro-organisms. Culturing is more sensitive than using smear microscopy, but is more time consuming and requires a higher level of infrastructure to perform (skilled technicians, facilities for media preparation) [1].

This method is used to diagnose active tuberculosis, as well as drug susceptibility. On average, diagnosis take 4 weeks to complete, with another 4-6 weeks to detect drug susceptibility [26].

- **Interferon Gamma Release Assays (IGRAs):** IGRAs are a type of blood test that measures a patients immune system's response to MTB. Practically this test works by taking a blood sample and mixing it with a specific substance. There are two widely used IGRAs: the T-SPOT TB test and the QuantiFERON TB Gold test.
This test is fast, but does require laboratory facilities. Furthermore, the test only tests for latent TB and is considered to be less accurate if the patient is HIV positive [27].
- **Tuberculin skin test:** The most popular tuberculin skin test is the Mantoux skin test. This method tests the hypersensitivity of a patient to a derived form of MTB by injecting a small amount into the patients arm. The patient needs to return to the clinic within 48-72 hours for results. This test is inexpensive to perform but interpretation is considered difficult, because various factors such as age, coexisting illnesses and immunological status can influence the results of the test [28]. Thus the effectiveness of the test relies on the level of medical expertise. Also, this test cannot distinguish between latent and active TB, but merely if the patient is infected with MTB.
- **Chest radiography:** Medical imaging techniques such as X-ray and computed tomography (CT) scans can be applied to the chest to view TB manifestations in the lungs. This method is effective and fast, but requires expensive equipment and experienced doctors to interpret the images. This method does not test for *extra-pulmonary* TB.
- **GeneXpert:** The GeneXpert test is a relatively new diagnosis method developed by the Foundation for Innovative New Diagnostics (FIND). This method is recommended by the WHO as a test for drug susceptibility and as an initial diagnostic tool, with some studies reporting specificity and sensitivities in the high 90's [29][30]. The GeneXpert tests use a process called polymerase chain reaction to identify and multiply the DNA sequence of the MTB within a sputum sample if present. Using this process, the test can also test for different drug resistances, as these show as mutations in the MTB DNA.

This method is comprehensive and effective, but unfortunately expensive. One test costs US\$9.98 and the associated equipment costs approximately US\$17,000. Additionally, GeneXpert machines require constant power to operate [26].

In summary, there are various existing tests for TB, with different advantages and disadvantages. Generally, the faster and more effective a test is, the higher the cost of the test. The GeneXpert test stands out as the most effective and reliable test, however it has a high operating cost. Currently, where clinics cannot afford expensive machines, more basic methods are used for diagnosis which results in a longer diagnosis period. Thus, investigating the possibility of a test that can be used to indicate the probability of a patient having TB within a short time frame is warranted if such a system could increase the cost effectiveness of low resource clinics.

1.5.5 Treatment of Tuberculosis

TB is a treatable and curable disease given the right medication and the infrastructure to monitor patient medication intake. The standard treatment for TB requires the patient to take antimicrobial drugs for a period of 6 months, during which he or she is under supervision to ensure consistent intake.

As standard treatment, four different first-line antimicrobial drugs are prescribed to account for the possibility of the patient having single drug resistant TB. However, in some cases, a patient can develop Multi-Drug Resistant TB (MDR-TB), usually caused by the incorrect or incomplete intake of first-line drugs such as Isoniazid and Rifampicin. Treating MDR-TB is much more difficult. Different drugs that are not always available and are much more expensive are required. A chemotherapy course of up to two years may be necessary, making the treatment more costly [18]. Therefore, ensuring a patient completes their TB medication course is crucial.

1.6 Summary

In this chapter the project objective of automatic diagnosis of TB through cough sound analysis was identified and placed in context. The scope of this project involves the design of a baseline system, able to distinguish between coughs of healthy patients and patients with TB, including the design of an automatic cough detection system from continuous audio. The human respiratory system, causes of coughing, the human auditory system, the pathology of TB and its symptoms were briefly discussed. Different currently available TB diagnosis methods were compared in terms of time, cost and efficiency and the problem of MDR-TB was mentioned.

Chapter 2

Literature Review

The analysis of coughing sounds has been an active area of research since 1989 with studies investigating the differences in acoustic and dynamic characteristics between different pulmonary diseases [31]. Generally cough sound analysis can be divided into two subgroups: Cough Detection and Cough Classification.

2.1 Cough Detection

The process of identifying, separating and counting coughs from recordings is a relatively well-researched field, with the first cough detection system proposed in 1964 [32]. However, no cough detection system is commercially available at this time [33].

Cough detection systems are useful for gathering information about cough frequency and cough intensity, which can give important insight into patient recovery as well as the severity of an illness. It is difficult to compare different studies in the field of cough detection, because no standard framework currently exists. Most studies try to distinguish cough sounds from other environmental sounds, such as dogs barking, speech and other respiratory sounds such as sneezing and snoring.

2.1.1 Specific Studies

In [33], the authors designed a cough detection algorithm that first distinguishes between voluntary coughs (coughs voluntarily produced at request) and the speech of healthy patients. This algorithm is then extended to distinguish between coughing and other sounds generated in the upper respiratory tract such as throat clearing, laughing and sneezing. Finally, the effectiveness of the algorithm in distinguishing between cough and non-cough sounds is tested on patients with respiratory diseases undergoing daily activities. The system was first unsuccessfully implemented using decision trees. Use of a neural network classifier improved the results significantly.

Recording was performed at a sampling rate of 44 kHz and a resolution of 16 bits per

2.1 Cough Detection

sample. For preprocessing of the recorded data, sound events were detected by using a non-overlapping sliding window (of unspecified length). The standard deviation of the amplitude was computed within each window, and then compared to an empirically determined threshold value. Sound event exclusion was performed on the basis of the length of the detected sound event. According to [33], the mean duration of a voluntary cough in their dataset is 0.3 ± 0.01 sec. Thus sound events significantly shorter than this duration were excluded from further analysis.

To distinguish between coughing and speech, the following features were used: The slope of the power spectral density was calculated around each maximum, as the slope of cough sounds show a sharp rise leading up to a maximum, as opposed to speech, where there is usually a more gradual increase. Additionally, the *Kurtosis* and *Skewness* of the spectrum was calculated for each cough event, as well as Mel Frequency Cepstral Coefficients (MFCC). It is unclear how the feature vectors were defined in this study, but from interpretation it seems each cough event was represented by a single feature vector.

In [33], the results achieved distinguishing between spontaneous cough and other non-cough sounds from patients infected with pulmonary illnesses were as follows:

Table 2.1: Classification Results presented in [33].

Classification Method	Sensitivity(%)	Specificity (%)
Decision Tree	28	99
Artificial Neural Network	82	96

A 24-h automated cough monitor (The Leicester Cough Monitor (LCM)) is described in [34]. This device is intended to measure cough frequency in a home environment. For this study, 6-hour recordings were collected from 15 patients with different illnesses characterised by chronic cough as a symptom, such as asthma, gastro-oesophageal reflux and eosinophilic bronchitis. The cough detection algorithm is based on the methods used successfully in speech recognition. A keyword spotting approach based on hidden Markov models is used to recognise sub-segments of a cough in the same way that phonemes are recognised in a speech recognition system.

For validation, the process was split into two phases. In the first phase, the cough counts of the LCM were compared to the results of manual sound analysis done on the 1st and 4th hours of recordings of 9 randomly selected patients. Thus 2 of the 6 hours of recorded data of 9 patients were used in this phase. For the manual analysis, 2 independent human labellers annotated the data (labeller 1 annotated the data twice and labeller 2 annotated it once). Only cough events that were in agreement by all three annotations were considered true. In the second stage, all the recorded data was used and manual annotation was done by only one observer on all 6 hours (from the study, it seems as though one

2.2 Cough Classification

observer listened to $6 \times 15 = 70$ hours of recorded data). In addition, another 50 patients with chronic cough were recorded for 24 hours. This data was not manually annotated, but used to determine the repeatability of the system. The LCM had a sensitivity and specificity of 86 and 99%, respectively, for detecting cough sounds. [34].

More recently, a cough frequency monitoring system has been designed for the monitoring of patient recovery from pulmonary tuberculosis [35]. As was the case in [34], the system described in [35] is designed to be a continuous cough counting system. However, in this study an event detection algorithm was implemented to reduce the amount of recorded data that needed to be subjected to manual analysis. This event detection algorithm identifies potential cough sounds by thresholding a smoothed energy measure for the signal. This process was effective in detecting cough events but not very accurate, frequently detecting non-cough events such as speech. Thus another step was introduced into this algorithm similar to the approach used in [33]. Instead of using the power spectral density, the slope of the signal energy was compared to a noise threshold. The noise threshold was determined by obtaining the 10th percentile of the signal energy within a 20 second sliding window.

MFCC features were used to train three different classifiers: a multilayer perceptron (MLP) neural network, a support vector machine (SVM) and sequential minimal optimization (SMO). The best performance was obtained using the MLP. However, for ease of implementation, the SMO classifier was used for classification as the difference in overall accuracy was not very big (88.2 % vs. 86.4%).

With a sensitivity of 81% this algorithm correctly detected most cough events with an average of 3.3 false alarms/hour (the actual average coughs per hour is not specified). Thus, by monitoring the cough frequency of patients recovering from TB, patients who are failing the treatment can be identified earlier and can be placed on a different type of treatment.

Various methods have been explored for data collection in cough detection research, including contact microphones (placed on the patient's thorax or trachea) [36], a free-field microphone necklace [34], hand-held recorders [35] and stationary microphones interfaced with a PC. There does not seem to be a clear advantage of one method over another, as different scenarios require different recording equipment.

2.2 Cough Classification

Cough classification aims to diagnose specific illnesses, or to classify cough types, through analysis of cough sounds. The cough sounds are usually recorded in controlled environments to minimize unwanted interfering noise, as opposed to cough detection, which is

2.2 Cough Classification

designed to perform in noisy environments.

The process of cough classification is based on the hypothesis that different pulmonary illnesses produce distinctively different cough patterns. According to [37], the spectral analysis of coughs gives insight into the timbre, or tonal quality, of different types of cough, which in turn can assist a diagnosis. For example, a cough with a brassy or bi-tonal timbre is such a strong characteristic of lymphoid gland tuberculosis that it may suffice as a diagnosis tool in itself [37]. The authors of [38] recently demonstrated that cough sounds carry distinctive spectral features, such as non-Gaussianity and Mel Cepstra, that can be used to diagnose pneumonia.

Cough classification can consist of various steps. In the following, the steps most commonly taken by various research studies are discussed.

2.2.1 Data Acquisition

Data acquisition for cough classification is usually performed in a closed and controlled setting in order to minimize environmental noise in the recording data. For the recording process, stationary microphones are set up in a clinic or hospital and recordings of patients are taken overnight. Alternatively, patients are taken to a recording chamber where cough sounds are explicitly recorded. Both these methods require the segmentation of cough events from the recording data. This is usually accomplished by a physician, meaning that it can be regarded as a manual, professional diagnosis. This will form the ground truth for subsequent classifier training.

In addition to recording data, other information regarding the patient can be of use, such as patient age, race, ethnicity, medical history and most importantly a list of current symptoms. The authors of [38] and [35] established that the inclusion of measurements other than recording data can increase the classification accuracy.

Once the data has been acquired, it is segmented into cough events. Each event can then itself be split up into windows (frames) varying in length from 10ms to 50ms [34][35][36].

2.2.2 Feature Extraction & Selection

Cough sounds can be divided into three phases: Inhalation, Forced Exhalation and Glottal Closure [8][39][36][40]. The second of these three phases has been identified by most studies as the definitive phase, containing most information regarding cough characteristics. Phase two begins when the amplitude has significantly reduced from its initial peak [35]. **Figure 2.1** shows an example waveform of a forced cough from a healthy patient indicating the three phases. This cough forms part of the dataset collected for this study at the Brooklyn Chest Health Clinic.

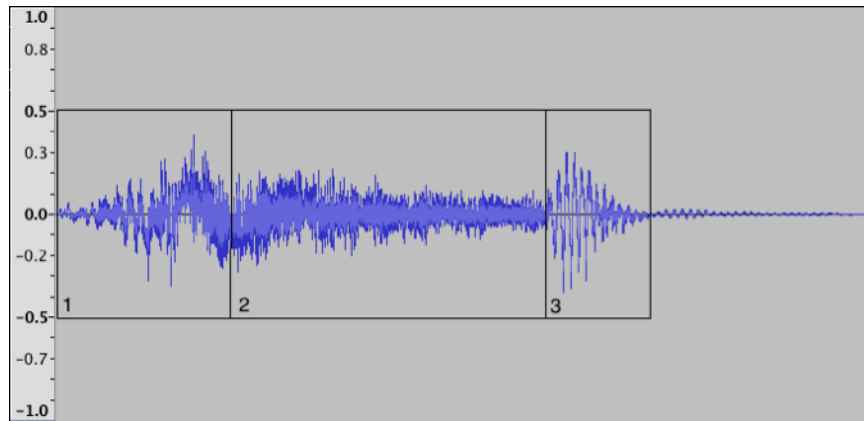


Figure 2.1: Example of a cough waveform with three phases indicated.

In cough classification, the sound waveform is analysed in the time and frequency domain by extracting features. The following features were used by the studies researched:

- Frequency domain

- Bispectrum Score (BGS) [38][2][7]

- Mel-frequency cepstral coefficients (MFCC)[38][2][7][33][8]

- Non-Gaussianity score (NGS)[38][2][7]

- Formant Frequencies (FF)[38][2]

- Spectral Kurtosis[38][2][7]

- Time domain

- Log energy[38][2][7]

- Zero Crossing Rate (ZCR)[38][2][7]

2.2.3 Specific Studies

In this section, specific studies relevant to cough classification will be discussed. Not much material on this topic could be identified.

One body of research tries to classify cough sounds as dry or wet. This is an important step in the diagnosis process of pulmonary illnesses, because it is currently very subjective. Usually a doctor would ask a patient to describe his or her cough in terms of dryness / wetness, or if possible, would ask the patient to produce a cough in order for the doctor to perform this classification. As mentioned in **Section 1.3.1**, the dryness / wetness of a cough is considered valuable information when narrowing down the list of possible illnesses contracted by the patient.

Three studies have focussed on dry/wet cough classification [8], [7], [40]. The studies in [8]

2.2 Cough Classification

and [40] used smaller datasets and used cough sounds from adults only. In [7], 178 cough events were used from 46 patients, while [8] used only 16 cough events from an unknown number of subjects and [40] used 30 cough samples from 10 different subjects. Also, [8] used only two features for classification. By extracting a wide range of features (63), and performing feature optimization to reduce the number of features to 23 (spectral and time domain), a mean sensitivity and specificity of $79 \pm 9\%$ and $72.7 \pm 8\%$ was achieved in [7]. Thus, an automated algorithm for the classification of dry and wet cough sounds has been proposed.

A second body of research considered an automated algorithm for the diagnosis of childhood pneumonia, first published as [38] and later updated in [2]. This work is relevant to what is proposed in our study, and is thus worth discussing in detail. We will first discuss the work in [38] and then compare it to the updated version in [2].

Using non-contact microphones for data acquisition, the algorithm involves the classification of cough sounds as *pneumonic* and *non-pneumonic*, indicating the presence - or absence of pneumonia in the patient respectively.

Figure 2.2 shows how the data was acquired. The recording environment usually consisted of single occupancy rooms in the respiratory ward of the hospital (Sardjito Hospital, Gadjah Mada University, Indonesia). However, occasionally the patient shared the room with another patient. Two microphones were used, one pointing towards the patient and one pointing in the opposite direction. The microphones were directed in this way to facilitate background noise subtraction in order to separate cough events from ambient sounds. The distance between the patients and the microphone directed at them varied between 40cm and 70cm.

The data was gathered from 91 patients (63 pneumonia and 28 non-pneumonia) and was divided into a *Model Development Dataset* (D_{MD}) and a *Prospective Validation Dataset* (D_{PV}). D_{MD} consisted of $N_{MD} = 66$ patients (46 pneumonia and 20 non-pneumonia), while D_{PV} consisted of $N_{PV} = 25$ patients (17 pneumonia and 8 non-pneumonia). For the non-pneumonia set, diseases such as asthma, bronchitis and pharyngitis are grouped together.

In most cases (more than 85 %), 5 - 10 coughs were obtained for each patient. Segmentation was done manually by a researcher unrelated to the study. The average duration of each recording in the dataset was 4h and 3 min with a standard deviation of 1h and 39 min.

For the feature extraction process, each cough event was divided into three non-overlapping frames and for each frame, the following features were computed: 12 Mel Frequency Cepstral Coefficients, Bispectrum Score (BGS), Log Energy, Non-Gaussianity Score (NGS),



Figure 2.2: Recording setup used for pneumonia cough classification in [38].

Zero Crossing Rate (ZCR), Spectral Kurtosis and the first four Formant Frequencies. Feature extraction was performed on all cough events in D_{MD} to create the feature set \mathbf{f}_{OC} , consisting of 63 different features in total.

Using \mathbf{f}_{OC} and a Leave-One-Out-Validation (LOOV) method, N_{MD} (66) logistic regression models (LRMs) were trained. The mean sensitivity and specificity for individual cough events are reported as $81 \pm 1\%$ on the *training set* and 63% and 52% on the *validation set*.

A smaller feature set was obtained by calculating the per-feature mean p-value in \mathbf{f}_{OC} over N_{MD} LRMs and then only including features with a p-value greater than a certain threshold, creating the selected feature set \mathbf{f}_{OS} . A new set of LRMs was designed using \mathbf{f}_{OS} , and the best LRM chosen using K-Means clustering by selecting the LRM with the lowest mean square error with respect to the cluster centroid. The mean *validation* sensitivity and specificity for individual cough events increased from and 63% and 52% to 69% and 64% respectively.

Once an optimal LRM was selected, the classification process was altered to perform a *diagnosis* for each patient instead of classifying each cough event. This was done by calculating a new metric called the Pneumonic Cough Index (PCI), which is the ratio of total number of coughs vs the number of coughs classified as pneumonic. The PCI was calculated as follows: For each patient, let P equal the total number of coughs while Q equals the number of coughs classified as pneumonic by the LRM. The Pneumonic Cough Index is then calculated as:

$$PCI = \frac{Q}{P} \quad (2.1)$$

2.2 Cough Classification

The PCI value was then compared to a threshold empirically selected as $\gamma_{PCI} = 0.5$ and a patient was diagnosed as pneumonic if $PCI \geq \gamma_{PCI}$. This method increased the *validation* sensitivity and specificity to 93% and 90.5% respectively.

The feature set \mathbf{f}_{OS} was then augmented with clinical measurements, including age, the presence of fever and the breathing rate index (BrI) (**Equation 2.2**, where BR refers to the measured breathing rate and age is measure in months) to create the augmented feature set \mathbf{F}_{AC} . The highest reported *validation* PCI based sensitivity and specificity are 95.6% and 90.5%.

$$BrI = \begin{cases} BR - 20 & \text{if Age} \geq 60 \text{ months} \\ BR - 40 & \text{otherwise} \end{cases} \quad (2.2)$$

Performing classification on the *Prospective Validation Dataset* (D_{PV}) using many different feature combinations (augmented, non-augmented, semi-augmented) and different LRM-feature set combinations, the optimal combination was found. The authors show that this system can diagnose pneumonia in patients with a sensitivity of greater than 90% and a specificity of greater than 85%. The study also states that the algorithm needs only 5 to 10 coughs per patient to reach these diagnosis results.

In [2], the researchers decided to improve on their previous study by training an artificial neural network (ANN) to perform classification. In this study however, classification was limited to pneumonia and asthma, whereas in [38] the non-pneumonia class contained various other pulmonary diseases.

In this study, a total of 18 patients were recorded (9 pneumonia and 9 asthma) with the recording setup as in the previous study. The age of the patients range from 1 - 86 months, with an average of 25 months. From these recordings, the first 50 coughs were manually selected from the continuous audio. These cough episodes were combined to form the complete dataset D . D consisted of a total of 674 coughs, of which 412 were from patients with pneumonia and 262 from patients with asthma.

Feature extraction was performed on each cough episode in D by first breaking up a cough into non-overlapping windows of 20ms. For each window, the following 22 features were computed to: 13 MFCCs, first five formant frequencies, ZCR, NGS and Shannon entropy. The feature extraction step was repeated for each window in each cough, constructing the feature vector matrix G .

For classification, an ANN with 1 input layer, 2 hidden layers and an output layer. The

number of neurons in each layer was chosen as 110, 20, 10 and 1 respectively. A linear activation function was used for the input layer and a sigmoid activation function was used for both hidden layers. A LOOV procedure was used so that each subject was used for testing exactly once.

The algorithm developed in this study has a reported sensitivity of 88.9% and specificity of 100%.

2.3 Summary

In this chapter, the literature applicable to this study was divided into two main fields: cough detection and cough classification. For each field, various studies were considered and their findings reported. For cough detection, a considerable amount of research has been done, with the best detection accuracy reported as 86.4% by [35]. For cough classification, research has been done on the classification between pneumonia and asthma in children and the classification between wet and dry coughs. However using cough classification for broader lung health diagnosis has yet to experience much attention.

Chapter 3

Classification and Evaluation

Methods

In this chapter, the different machine learning algorithms used in the study are discussed. Logistic regression, hidden Markov models (HMMs) and Gaussian mixture models (GMMs) are discussed. Decision trees (DT) are discussed in less detail (in **Section 6.4.2**) as the results obtained with DT were sub optimal. Furthermore, the different evaluation methods and metrics used are discussed.

3.1 Hidden Markov Models (HMMs)

HMMs are a popular machine learning algorithm in fields where sequential data are to be modelled and classified, such as in speech recognition and DNA sequencing. Thus, using HMMs is a good choice when aiming to classify acoustic events in a recording, as is the goal in automatic annotation.

Hidden Markov models (HMMs) are an extension of Markov models, where each observation is a function of the current state, and each state is directly influenced only by the previous state. Before considering hidden Markov models, we will introduce Markov models. The following section is inspired by the developments in [41], [42] and [43].

3.1.1 Markov Models

Markov models (or Markov chains) are a way of representing a sequence of successive states at a certain time t . The number of states in a Markov model is finite and can thus be seen as a type of finite state machine. In contrast to a finite state machine, the transitions between states are governed by probabilities and not explicit events.

If we define a set of states $S = \{s_1, s_2, \dots, s_{|S|}\}$ with $|S|$ representing the number of states,

3.1 Hidden Markov Models (HMMs)

we can denote any state at a certain time t as $s_i(t)$. The states in a Markov model can occur in a sequence of T states \mathbf{s}^T . A specific sequence can thus be denoted as $\mathbf{s}^6 = \{s_1, s_4, s_2, s_2, s_3, s_4\}$, with $|S| = 4$ and $T = 6$ for this example.

A model is described by its *transition probabilities* $P(s_j(t+1)|s_i(t)) = a_{ij}$, which defines the probability of moving from state i to state j . Transition probabilities do not have to be symmetric and one state may be visited repeatedly. **Figure 3.1** shows a three state Markov model with transition probabilities defined by a_{ij} .

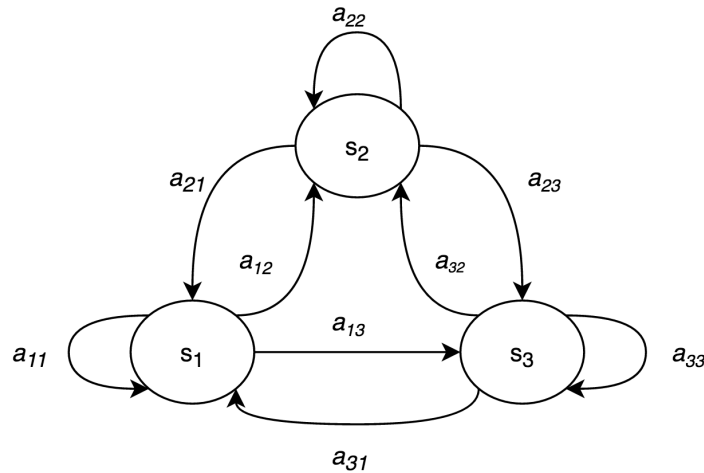


Figure 3.1: A three state Markov model.

Given a complete transition probability matrix, it is possible to compute the probability that a model has produced a particular state sequence. For instance, given the transition matrix θ we can determine the probability that the model produced the state sequence \mathbf{s}^T , defined above, using:

$$P(\mathbf{s}^6|\theta) = a_{14}a_{42}a_{22}a_{23}a_{34}$$

This leads to a formal definition: The probability of a specific state sequence is the sum of all the applicable transition probabilities, or:

$$\begin{aligned}
 P(\mathbf{s}^T|\theta) &= \sum_{t=1}^{t=T} a_{ij} \\
 &= \sum_{t=1}^{t=T} P(s_j(t+1)|s_i(t))
 \end{aligned} \tag{3.1}$$

3.1.2 Hidden Markov Models

One shortcoming of a Markov model is that in many problems one is unable to externally observe the current state. It is only possible observe a probabilistic function of each state.

3.1 Hidden Markov Models (HMMs)

To better explain this, consider the following example by Jason Eisner [44]:

You are climatologists in the year 2799, studying the history of global warming. You can't find any records of Baltimore weather, but you do find my (Jason Eisner's) diary, in which I assiduously recorded how much ice cream I ate each day. *What can you figure out from this about the weather that summer?*

In this example, the sequence of states would be the weather of each day, and the observed data is the amount of ice cream consumed by the author of the diary. Here we can see that the states are not explicitly observable, but it seems as though it should be possible to make informed decisions as to what the weather was each day by considering the observed ice cream consumption.

Because the states that the Markov model enters are unobservable, the name *Hidden Markov Model* arises.

To state this formally, a hidden Markov model is a Markov model for which we can observe some emitted value or symbol $x(t)$ from a set of possible observed values $V = \{v_1, v_2, \dots, v_{|V|}\}$ ($x \in V$) for each state $s(t)$ at time t .¹ **Figure 3.2** shows the a three state HMM with emission probabilities defined as b_{jk} and observable values defined as x_k .

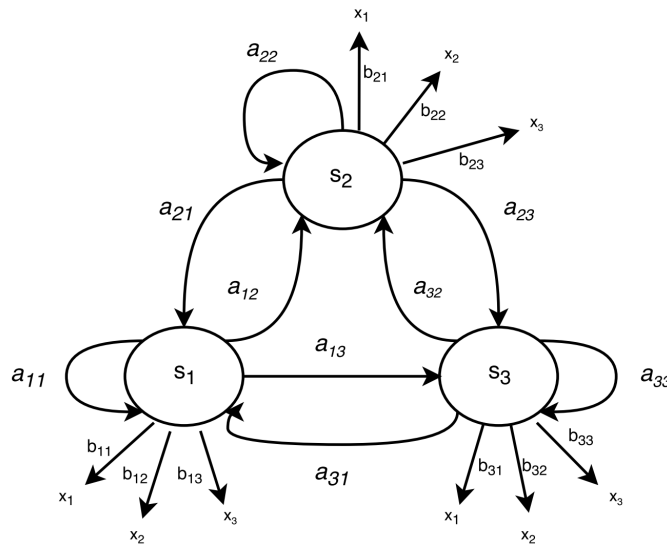


Figure 3.2: Hidden Markov model with 3 states $\{s_1, s_2, s_3\}$ and three observable values $\{x_1, x_2, x_3\}$.

In the same way we defined a sequence of states as \mathbf{s}^T with T the number of visited states, we can define a sequence of observed visible symbols as $\mathbf{X}^T = \{x(1), x(2), \dots, x(T)\}$. Thus we can have a particular sequence of observed values $\mathbf{X}^6 = \{x_5, x_1, x_1, x_5, x_2, x_3\}$.

¹For this introduction, we will restrict the nature of the observed output to discrete symbols. However, it can be generalised to continuous observations.

3.1 Hidden Markov Models (HMMs)

Now define the probability of emitting value $x_k(t)$ when the model is in state $s_j(t)$ as $P(x_k(t)|s_j(t)) = b_{jk}$.

To summarize, we have:

$$\left. \begin{aligned} a_{ij} &= P(s_j(t+1)|s_i(t)) \\ b_{jk} &= P(x_k(t)|s_j(t)) \end{aligned} \right\} i, j \in 1..c \text{ and } c = \# \text{ states} \quad (3.2)$$

Also, it is important to note that:

$$\begin{aligned} \sum_j a_{ij} &= 1 \text{ for all } i \\ \sum_j b_{jk} &= 1 \text{ for all } k \end{aligned}$$

which states that the sum of the probabilities of all outgoing links must be equal to 1, and the sum of all observation probabilities must also be equal to 1.

In the optimization of an HMM, we generally focus on three tasks, namely:

- (a) To determine the probability of an observed sequence \mathbf{X}^T being emitted by a model, given the transition and emission probabilities a_{ij} and b_{jk} .
- (b) Given a specific observation sequence \mathbf{X}^T , emitted by a defined HMM, determine the most likely sequence of *hidden* states \mathbf{s}^T to generate \mathbf{X}^T .
- (c) To estimate the values for all a_{ij} and b_{jk} , using a defined model structure (number of hidden states and possible observations) and a set of training data (observations with known hidden state sequences).

(a) Probability of observed sequence

The probability that a defined HMM produces a specific sequence of observations \mathbf{X}^T can be calculated using:

$$P(\mathbf{X}^T) = \sum_{r=1}^{r_{max}} P(\mathbf{X}^T | \mathbf{s}_r^T) P(\mathbf{s}_r^T) \quad (3.3)$$

with each r referring to a unique hidden state sequence $\mathbf{s}_r^T = \{w(1), w(2), \dots, w(T)\}$ with T the number of hidden states. **Equation 3.3** states that in order to compute the probability of a specific observation sequence, we need to compute the probability that \mathbf{X}^T was produced by each possible hidden state sequence and sum all these probabilities.

3.1 Hidden Markov Models (HMMs)

The terms in the sum of **Equation 3.3** can be simplified. The second term in the sum can be rewritten, by using **Equation 3.1** for the each specific state sequence indexed by r and at time t , with respect to $t - 1$:

$$P(\mathbf{s}_r^T) = \sum_{t=1}^{t=T} P(s_j(t)|s_i(t-1)) \quad (3.4)$$

The first term in the sum of **Equation 3.3** can also be rewritten, because of our assumption that the observation probabilities are only dependent on the current state:

$$P(\mathbf{X}^T|\mathbf{s}_r^T) = \prod_{t=1}^T P(x(t)|s(t)), \quad (3.5)$$

which translates to take the product of all the observation probabilities b_{jk} for a hidden state sequence indexed by r .

Substituting **Equations 3.4** and **3.5** into **3.3** we get:

$$P(\mathbf{X}^T) = \sum_{r=1}^{r_{max}} \prod_{t=1}^T P(x(t)|s(t))P(s_j(t)|s_i(t-1)) \quad (3.6)$$

Equation (3.6) has an intuitive interpretation: To calculate the probability that a model has produced a specific observation sequence \mathbf{X}^T , we need to sum over r_{max} possible hidden state sequences and multiply each term in the sum by the probability that it has emitted each observation in our observation sequence. All this information is available in the transition probabilities a_{ij} and emission probabilities b_{jk} .

However, direct application of **Equation (3.6)** is very computationally expensive, with an order of complexity of $O(c^T T)$. For example, if we have a model with 5 states ($c = 5$) and we want to determine the probability of an observation sequence of length 15 ($T = 15$), it would require $O = 5^{15} \times 15 = 457763671875$, or roughly 457 billion calculations.

To reduce the computational complexity of **Equation (3.6)**, it can be reformulated into what is referred to as the *Forward Algorithm*.

First we define:

$$\alpha_i(t) = \begin{cases} 0 & t = 0 \text{ and } i \neq \text{initial state} \\ 1 & t = 1 \text{ and } i = \text{initial state} \\ \sum_j \alpha_j(t-1)a_{ji}b_{jk}x(t) & \text{otherwise,} \end{cases} \quad (3.7)$$

with $b_{jk}x(t)$ denoting the emission probability b_{jk} limited to the observation $x(t)$ at time t , which will be the only non-zero term in the sum at index k matching the emitted ob-

3.1 Hidden Markov Models (HMMs)

ervation $x(t)$.

Note that $\alpha_j(t)$ represents the probability that our model is in state s_j at time t after emitting the first t observations in \mathbf{X}^T .

Subsequently, we can define the *Forward Algorithm* as a recursive implementation of **Equation (3.6)**:

Algorithm 1 Forward Algorithm

```

begin
  initialize:  $s(1), t = 0, a_{ji}, b_{jk}$ , observed sequence  $\mathbf{X}^T, \alpha(0) = 1$ 
  for  $t \leftarrow t + 1$  until  $t = T$ 
     $\alpha_j(t) \leftarrow \sum_{i=1}^c \alpha_i(t-1) a_{ji} b_{jk} x(t)$ 
  end for
  return  $P(\mathbf{X}^T) \leftarrow \alpha_0(T)$  for the final state
end

```

The forward algorithm sums over c terms T times, thus it has a computational complexity of $O(c^2T)$, which is a substantial improvement over **Equation (3.6)**. If we take the same example used before, with $c = 5$ and $T = 15$, the forward algorithm will perform just $O = 5^2 \times 15 = 375$ operations.

(b) Viterbi Algorithm

Now that we can determine the probability of a certain observed sequence, next we are interested in computing the most likely sequence of hidden states given an observed sequence. Thus, we want to determine the most probable \mathbf{s}^T that would have generated a certain \mathbf{X}^T .

One way of calculating this is by enumerating all possible state sequences that could have produced \mathbf{X}^T , and taking the path with the highest probability using **Equation (3.1)**. However, this would result in computational complexity of $O(c^T T)$ which is impractical. A more efficient way of doing this is by using the *Viterbi Algorithm*, shown in **Algorithm 2**, which can be implemented with a computational complexity of $O(c^2T)$.

The Viterbi algorithm determines the hidden state sequence from the observed values \mathbf{X}^T by recursively calculating the values of $\alpha_j(t)$ (the probability of being in state j after observing t values in \mathbf{X}) for each t using only the highest value of $\alpha_{j'}(t-1)$. Thus, by starting with $\alpha(0) = 1$, which means we know which state the system is in at time $t = 0$, for each consecutive observed value in \mathbf{X}^T , compute the values of $\alpha_j(t)$ using $\alpha_{j'}(t-1)$ for all possible states s_j with $j = 1, \dots, |S|$. Then set $\alpha_{j'}(t)$ to the highest value of $\alpha_j(t)$ and save state j' to the hidden state path. This is repeated for all observed values in \mathbf{X}^T . Thus, the hidden state path is determined by recursively computing the probability of

3.1 Hidden Markov Models (HMMs)

being in a certain state, while only considering the previous state with the highest probability.

Algorithm 2 Viterbi Algorithm

```

begin
  initialize: Path  $\leftarrow \{\}$ ,  $t \leftarrow 0$ 
  for  $t = 0 \rightarrow T$ 
    set  $k = 0, \alpha(0) = 1$ 
    for  $k = 0 \rightarrow c$ :
       $\alpha_k(t) \leftarrow b_{jk}x(t) \sum_{i=1}^c \alpha_i(t-1)a_{ji}$ 
    end for
     $j' \leftarrow \arg \max_j \alpha_j(t)$ 
    AppendTo Path  $s_{j'}$ 
  end for
  return Path
end
```

(c) Baum-Welch Estimation

We can now determine some important aspects of an HMM if we are given its state transition probabilities a_{ij} and output emission probabilities b_{jk} . The final task is to determine these parameters by learning from known examples. This process is generally referred to as *training* the HMM and is realized by performing a process called the *forward-backward algorithm* or *Baum-Welch estimation*.

In **Equation (3.7)** we defined $\alpha_i(t)$, which is the probability that the model is in state s_i at time t and has emitted the correct visible outputs up to this time.

Let us further define:

$$\beta_i(t) = \begin{cases} 0 & s_i(t) \neq \text{final state of sequence and } t = T \\ 1 & s_i(t) = \text{final state of sequence and } t = T \\ \sum_j \beta_j(t+1)a_{ij}b_{jk}x(t+1) & \text{otherwise,} \end{cases} \quad (3.8)$$

where $\beta_i(t)$ is the probability that the model is currently in state $s_i(t)$ and *will emit* the remainder of the output sequence in \mathbf{X}^T , from $t+1 \rightarrow T$.

In order to determine the values of β_j for all time $t = 0 \rightarrow T$, we will follow the time reversed procedure of **Algorithm 1**. Thus follows the definition of the *Backward algorithm*, shown in **Algorithm 3**.

In the last line of the algorithm, the probability of the model emitting the observed sequence is now set to the value of $\beta_j(0)$. This makes sense, because $\beta_j(0)$ is the probability that the model is in state $s_j(0)$ and that the system will produce the remainder of the observed sequence $P(\mathbf{X}^T)$, ie the probability that the model will emit the entire

3.1 Hidden Markov Models (HMMs)

Algorithm 3 Backward Algorithm

```

begin
  initialize  $s(t), t = T, a_{ji}, b_{jk}$ , observed sequence  $\mathbf{X}^T$ 
  for  $t \leftarrow t - 1$  until  $t = 1$ 
     $\beta_j(t) \leftarrow \sum_{i=1}^c \beta_i(t+1) a_{ji} b_{jk} x(t+1)$ 
  end for
  return  $P(\mathbf{X}^T) \leftarrow \beta_j(0)$ 
end

```

observed sequence when in its initial state.

Thus, with our values $\alpha_i(t)$ and $\beta_i(t)$, we have a way of measuring how well our model is performing up to time t and how well it will perform given everything up to time T .

The goal is now to use the values of $\alpha_i(t)$ and $\beta_i(t)$, calculated on some estimates of a_{ij} and b_{jk} , to determine the updated parameters \hat{a}_{ij} and \hat{b}_{jk} that better explain the data. First, define a new transition probability $\gamma_{ij}(t)$ - the probability that the model will move from state $s_i(t-1)$ to state $s_j(t)$ given that the model has produced the entire observed sequence \mathbf{X}^T through any hidden state path:

$$\gamma_{ij}(t) = \frac{\alpha_i(t-1) a_{ij} b_{jk} \beta_j(t)}{P(\mathbf{X}^T | \boldsymbol{\theta})} \quad (3.9)$$

where $P(\mathbf{X}^T | \boldsymbol{\theta})$ being the probability that the model has produced \mathbf{X}^T for any state path, given the probabilities $\boldsymbol{\theta} = \{a_{ij}, b_{jk}\}$.

By noting that the number of expected transitions from state $s_i(t-1)$ to $s_j(t)$ is equal to $\sum_{t=1}^T \gamma_{ij}(t)$, and at time step t , the number of expected transitions from state s_i to any state is equal to $\sum_{t=1}^T \sum_k \gamma_{ik}$, we can calculate updated values for the transition probabilities a_{ij} using the ratio of these two quantities:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^T \gamma_{ij}(t)}{\sum_{t=1}^T \sum_k \gamma_{ik}} \quad (3.10)$$

Following the same reasoning, we can calculate an updated estimate for b_{jk} by taking the ratio of the frequency that a specific observed value x_k is emitted over the frequency that any symbol is emitted:

$$\hat{b}_{jk} = \frac{\sum_{t=1}^T b_{jk} x(t)}{\sum_{t=1}^T b_{jk}} \quad (3.11)$$

Using **Equations 3.10** and **3.11**, the Baum-Welch algorithm updates the values of a_{ij} and b_{jk} recursively, until some convergence criteria is met.

Thus, using **Algorithm 4** we can iteratively train an HMM by optimizing the transition and emission probabilities using known output sequences.

Algorithm 4 Baum-Welch Algorithm

```

begin
  initialize  $a_{ij}, b_{jk}$ , training sequence  $\mathbf{X}^T$ , convergence criterion  $\theta$ 
  do  $z \leftarrow z + 1$ 
    compute  $\hat{a}(z)$  from  $a(z-1)$  and  $b(z-1)$  using Eq (3.10)
    compute  $\hat{b}(z)$  from  $a(z-1)$  and  $b(z-e)$  using Eq (3.11)
     $a_{ij}(z) \leftarrow \hat{a}_{ij}(z-1)$ 
     $b_{jk}(z) \leftarrow \hat{b}_{jk}(z-1)$ 
  until  $\max_{i,j,k} [a_{ij}(z) - a_{ij}(z-1), b_{jk}(z) - b_{jk}(z-1)] < \theta$ ;    (convergence)
  return  $a_{ij} \leftarrow a_{ij}(z)$ ;  $b_{jk} \leftarrow b_{jk}(z)$ 
end

```

3.2 Logistic Regression

Logistic regression is a generalized linear model, which can be used as a classifier by estimating the probability that an example belongs to a specific class. It centres around the use of the logistic sigmoid function, shown in **Figure 3.3**.

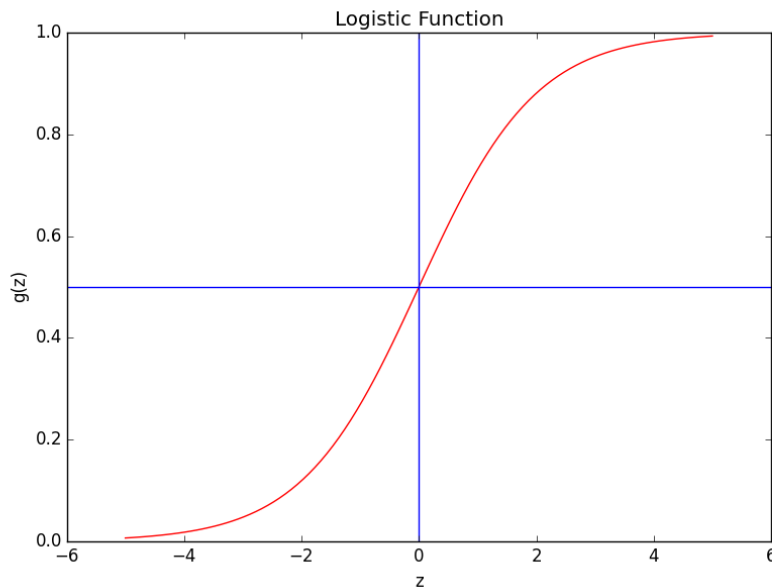


Figure 3.3: Logistic sigmoid function

With $\{x = x_0, x_1, \dots, x_n\}$ as the feature vector that must be classified, the probability that the vector belongs to a certain class is estimated using the hypothesis function:

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}} \quad (3.12)$$

where

$$\begin{aligned}\theta^T x &= \sum_{i=0}^n \theta_i x_i \\ &= \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n\end{aligned}$$

where n is the dimensionality of the input vector and $x_0 = 1$.

Now consider a binary classification problem¹ where the probability of $Y = 1$ can be estimated as:

$$P(Y = 1|X = x; \theta) = \frac{1}{1 + e^{-\theta^T x}} \quad (3.13)$$

$Y = 1$ represents the case when a cough event belongs to a patient that is TB-positive, and conversely $Y = 0$ indicates a cough event belonging to a TB-negative patient.

This means the probability that a given example belongs to class 0 or 1 can be estimated as $h_\theta(x)$ using the logistic function in (3.12). However before this can be done, the regression coefficients (also known as weight factors) θ need to be determined.

Ideally, we want to choose θ in order to maximise the probability that an unknown example is correctly classified. The process used to determine these values of θ is called **maximum likelihood estimation** and is described as follows:

First, assume that:

$$P(y = 1|x; \theta) = h_\theta(x) \quad (3.14)$$

$$P(y = 0|x; \theta) = 1 - h_\theta(x) \quad (3.15)$$

which can be written more concisely as

$$P(y|x; \theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

Assuming our input data X is independent and identically distributed (IID), we can obtain the probability of all the training data by taking the product of all the individually

¹Note that logistic regression is not limited to a binary classification problem and can be extended to multi-class classification by using a 1 vs all training method. However, this method is not used in this study and will not be discussed further.

calculated probabilities. Thus, we need to maximize the quantity:

$$\begin{aligned} L(\theta) &= \prod_{i=1}^n P(y|x; \theta) \\ &= \prod_{i=1}^n (h_{\theta}(x^{(i)})^{y^{(i)}} (1 - h_{\theta}(x^{(i)}))^{1-y^{(i)}}) \end{aligned}$$

By taking the logarithm, the product becomes a sum, and we can maximize the log likelihood:

$$\begin{aligned} l(\theta) &= \log(L(\theta)) \\ &= \sum_{i=1}^n y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \end{aligned} \quad (3.16)$$

where n is the number of samples.

To maximize the likelihood, we can use an iterative algorithm called batch gradient ascent. Batch refers to the fact that all values in the input vector x are used to update θ . Another version of this algorithm is called stochastic gradient ascent, which uses a small subset of the training data (also referred to as mini-batches) to update θ . Stochastic gradient ascent is better suited for when the hypothesis function has various local maxima. In the case of logistic regression, $h_{\theta}(x)$ is convex and does not have any local maxima, thus batch gradient ascent is an appropriate choice.

Batch gradient ascent starts by assuming a randomly chosen initial set of values for θ and then updating these values iteratively using the vectorized expression in (3.17) until the values converge, which indicates that the log likelihood $l(\theta)$ has reached a maximum. Here α represents the learning rate, or the rate at which values are updated, and $\nabla_{\theta} l(\theta)$ refers to the partial derivative of $l(\theta)$ with respect to θ

$$\theta := \theta + \alpha \nabla_{\theta} l(\theta) \quad (3.17)$$

To obtain an expression for $\nabla_{\theta} l(\theta)$, let $z = \theta^T x$ so that (3.12) becomes $g(z) = \frac{1}{1 + \exp^{-z}}$. Now

$$\begin{aligned} g'(z) &= \frac{d}{dz} \frac{1}{1 + e^{-z}} \\ &= \frac{1}{(1 + e^{-z})^2} (e^{-z}) \\ &= \frac{1}{(1 + e^{-z})} \cdot \left(1 - \frac{1}{1 + e^{-z}}\right) \\ &= g(z)(1 - g(z)) \end{aligned} \quad (3.18)$$

Recall that

$$\frac{d}{dx} \log(x) = \frac{1}{x}$$

and apply (3.18) to obtain the derivative of $l(\theta)$ for a single input and output (x, y) (thus ignoring the sum):

$$\begin{aligned} \frac{\partial}{\partial \theta_j} l(\theta) &= \left(\frac{y}{h_\theta(x)} + \frac{(1-y)}{1-h_\theta(x)} \right) \frac{\partial}{\partial \theta_j} h_\theta(x) \\ &= \left(\frac{y}{g(\theta^T x)} + \frac{(1-y)}{1-g(\theta^T x)} \right) g(\theta^T x)(1-g(\theta^T x)) \frac{\partial}{\partial \theta_j} \theta^T x \\ &= (y(1-g(\theta^T x)) - (1-y)g(\theta^T x)) x_j \\ &= (y - y(g(\theta^T x)) - g(\theta^T x) + y(g(\theta^T x))) x_j \\ &= (y - h_\theta(x)) x_j \end{aligned}$$

Now we can re-write **Equation 3.17** as follows:

$$\theta_j := \theta_j + \alpha (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)} \quad (3.19)$$

Equation 3.19 tells us that each weight factor θ_j will be updated with a value proportional to the error term $(y^{(i)} - h_\theta(x^{(i)}))$. Thus if the predicted value $h_\theta(x^{(i)})$ for training example $x^{(i)}$ is very close to the correct value of $y^{(i)}$, then the value of θ_j will not be updated much as a result of that training example.

When maximizing the loss function in **Equation 3.16** we need to account for the possibility of overfitting to the training data. Overfitting occurs when a model is optimized with no constraints in terms of model complexity and results in bad generalization. Intuitively, this means the model memorizes the training data, but cannot perform classification on unseen data.

Regularization penalizes model complexity by adjusting the weights assigned to each feature [45]. Different types of regularization methods are applicable to different learning algorithms. **Equation 3.20** shows one type of regularization called L2 regularization, which is frequently used with logistic regression .

$$\frac{\lambda}{2} \|\theta\|^2 = \frac{\lambda}{2} \sum_{j=1}^n \theta_j^2 \quad (3.20)$$

where λ represents the regularization coefficient.

By subtracting the regularization term from the log-likelihood function in **Equation**

3.16, we get the regularized log-likelihood $\hat{l}(\theta)$:

$$\begin{aligned}\hat{l}(\theta) &= l(\theta) - \frac{\lambda}{2} \|\theta\|^2 \\ &= \sum_{i=1}^n y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) - \frac{\lambda}{2} \|\theta\|^2\end{aligned}\quad (3.21)$$

Figure 3.4 shows a visualization of how L2 regularization affects the log-likelihood. The regularization term limits the possible weight values of θ to stay inside the circle and thus stops the log likelihood from reaching the unregularized optimum, which prevents the model from overfitting to the training data.

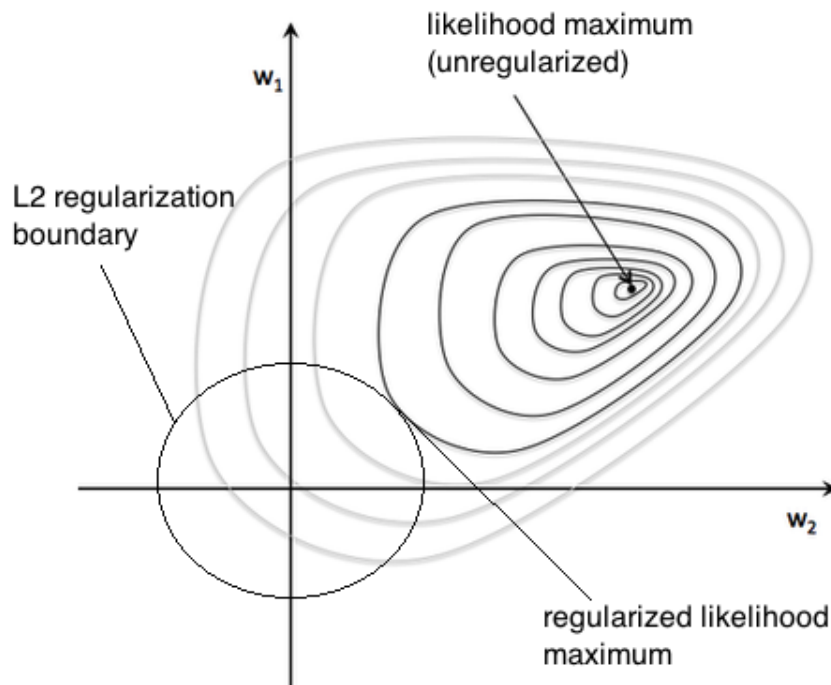


Figure 3.4: L2 regularization visualization modified from [45].

In **Equation 3.14** we defined $P(Y = 1|x; \theta) = h(x)$. This means that the probability that example x belongs to class 1 is equal to the logistic function $h_{\theta}(x)$ parametrised by θ . Thus, we can simply predict the class of input x by stating:

$$\text{prediction} = \begin{cases} 1 & \text{if } h(x) \geq 0.5 \\ 0 & \text{if } h(x) < 0.5 \end{cases}$$

However, if the prior probability of x belonging to class 1 ($P(Y = 1|X; \theta)$) is not 50%, choosing the threshold probability as 0.5 will not yield the best possible specificity and sensitivity.

3.3 Gaussian Mixture Models

One way of calculating the optimal threshold probability (later referred to as γ) for balanced specificity and sensitivity, is by using Receiver Operator Characteristic (ROC) curve analysis. This procedure is described in **Section 3.4.3**.

3.3 Gaussian Mixture Models

A Gaussian mixture model (GMM) is a probability density function parametrised as a weighted sum of Gaussian (normal) probability densities. GMMs can be used for supervised (classification) and unsupervised (clustering) learning.

A single Gaussian distribution is shown in **Figure 3.5** and is described by **Equation 3.22**, with σ and μ representing the standard deviation and mean of the distribution respectively (0.01 and 0 in this case).

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.22)$$

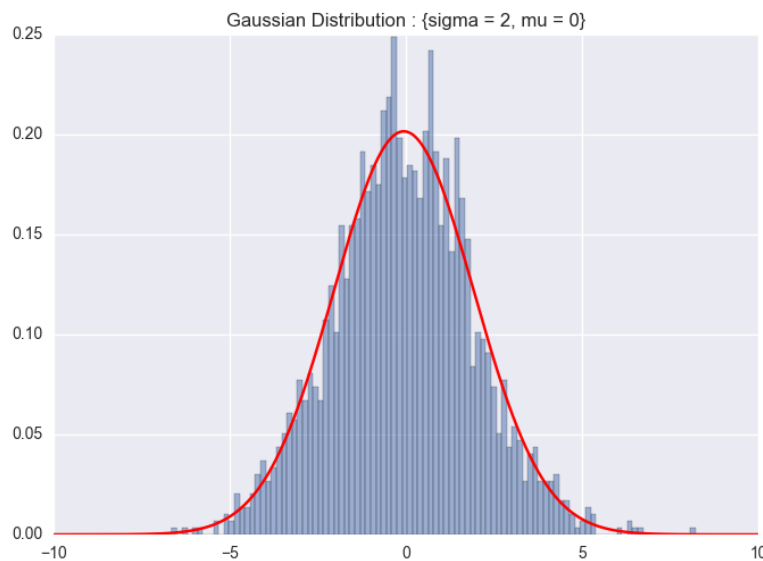


Figure 3.5: Single Gaussian distribution

A Gaussian distribution can be used to model a dataset. By computing the mean and standard deviation of the dataset, the probability of each data point can be determined using **Equation 3.22**. However, when modelling multi modal data (data that can be grouped into distinct groups) a single Gaussian distribution has serious limitations. This is illustrated in **Figure 3.6**. The data is characterized by two normal distributions, with means of 0 and 6 and variances of 3 and 0.5 respectively. A single mixture Gaussian distribution fails to model the data, while a GMM with 2 mixture components successfully models each group as a separate distribution.

3.3 Gaussian Mixture Models

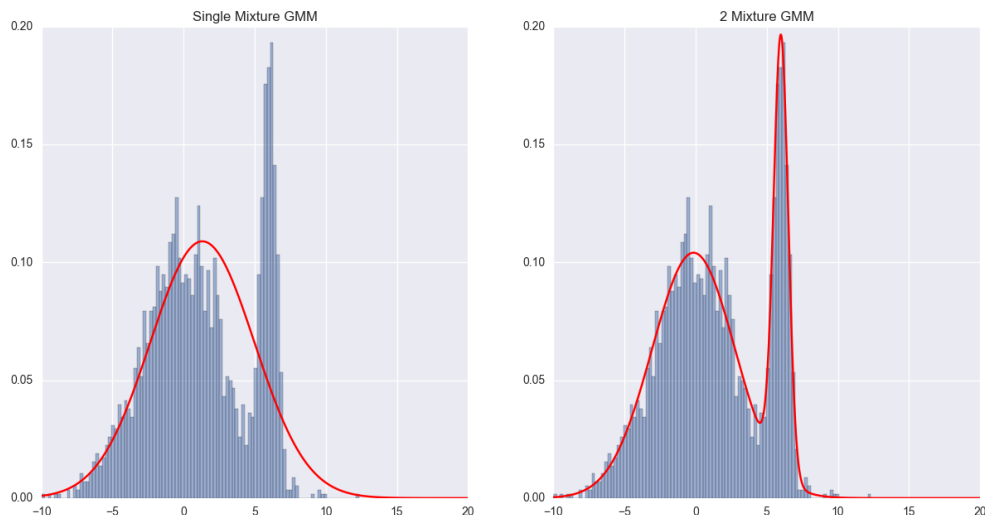


Figure 3.6: Example of multi modal Gaussian distributions.

A GMM is a weighted sum of multivariate Gaussian densities:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^M w_i g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \quad (3.23)$$

where w_i are the mixture weights with $i = \{1.., M\}$, \mathbf{x} is a D -dimensional data vector of continuous measurements and $g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ represents a multivariate Gaussian distribution which can be expressed as:

$$g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{(2\pi)^{D/2} |\boldsymbol{\Sigma}_i|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_i)' \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i) \right\}, \quad (3.24)$$

where $\boldsymbol{\mu}$ is the mean vector, $\boldsymbol{\Sigma}_i$ is the $D \times D$ covariance matrix and $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix.

The mixture weights w_i represent the prior probability of a sample belonging to mixture i and are bounded by the following constraints: $0 \leq w_i \leq 1$ and $\sum_{i=1}^M w_i = 1$.

The GMM parameters are collectively referred to using the notation:

$$\lambda = \{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}, \text{ with } i = \{1.., M\} \quad (3.25)$$

The optimal values of λ can be determined using the maximum-likelihood (ML) estimation method. This method aims to find parameters which maximize the likelihood of the GMM, given the training dataset. In other words, it tries to find the parameters that will make the observed data (the training data) the most probable.

Formally, given a dataset X of K training vectors: $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_K\}$, and assum-

3.3 Gaussian Mixture Models

ing each training vector is independent, the likelihood with respect to a GMM with parameters λ can be expressed as:

$$p(X|\lambda) = \prod_{k=1}^K p(\mathbf{x}_k|\lambda). \quad (3.26)$$

Using **Equation 3.23** to substitute $p(\mathbf{x}_k|\lambda)$, **Equation 3.26** can be written in log-form to convert the likelihood to a log-likelihood, :

$$\log(p(X|\lambda)) = \sum_{k=1}^K \log \left\{ \sum_{i=1}^M w_i g(\mathbf{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right\} \quad (3.27)$$

Unfortunately, because the log-likelihood $\log(p(X|\lambda))$ is a non-linear function of λ , there is no closed-form solution to the optimal values of λ . However, any iterative maximization algorithm such as gradient descent can be used to find these values. One such a maximization algorithm that has been established as computationally efficient for GMMs is the Expectation Maximization (EM) algorithm.

Given some initial model parameters λ , the EM algorithm iteratively estimates new model parameters $\bar{\lambda}$ such that $p(X|\bar{\lambda}) \geq p(X|\lambda)$. The parameters $\bar{\lambda}$ are then set as the new starting parameters λ and the process is repeated, until some convergence criteria is met.

The EM algorithm is dependant on the initial parameters and will converge to a local optimum, which is not ideal. Thus, there are several methods used for choosing the starting model parameters λ . One method is to choose random parameters, but this often leads to sub-optimal results. Another method is to use a clustering algorithm such as K-means to estimate the initial means and covariance values from the training set. Then the mixture weights w_i can be calculated using the fraction of samples assigned to each cluster. In other words, given a total of N samples, if cluster i , with mean and covariance μ_i and σ_i has n samples in that cluster, the mixture weight of that cluster can be computed as $w_i = n/N$.

The EM algorithm consists of two steps: an estimation step and a maximization step. In the estimation step, the posterior probability $P(i|\mathbf{x}_k, \lambda)$ for each mixture component is calculated using **Equation 3.28**. This probability represents the likelihood that the observation x_k belongs to mixture i .

$$P(i|\mathbf{x}_k, \lambda) = \frac{w_i g(\mathbf{x}_k|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)}{\sum_{j=1}^M w_j g(\mathbf{x}_k|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \quad (3.28)$$

In the maximization step, the posterior probability $P(i|\mathbf{x}_k, \lambda)$, calculated in the estimation step, is used to calculate new values for w , $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. By letting $P(i|\mathbf{x}_k, \lambda) = \gamma_i$,

these new values are calculated with the following equations [46]:

$$\bar{w}_i = \frac{1}{K} \sum_{k=1}^K \gamma_i \quad (3.29a)$$

$$\bar{\boldsymbol{\mu}}_i = \frac{\sum_{k=1}^K \gamma_i \mathbf{x}_k}{\sum_{k=1}^K \gamma_i} \quad (3.29b)$$

$$\bar{\Sigma}_i = \frac{\sum_{k=1}^K \gamma_i (\mathbf{x}_k - \bar{\boldsymbol{\mu}}_i)(\mathbf{x}_k - \bar{\boldsymbol{\mu}}_i)}{\sum_{k=1}^K \gamma_i} \quad (3.29c)$$

After updating the values of w , μ_i and Σ_i , the log-likelihood in **Equation 3.27** can be calculated again and the process can be iterated until some convergence criteria is met.

Once the optimal parameters have been determined, the GMM can be used as a classifier by training a GMM for each class in the training set. Subsequently, with λ set to the optimal parameters, the likelihood that a new sample vector \mathbf{x} belongs to the GMM of each class can be calculated using **Equation 3.23**. The sample is then classified as the class with the highest likelihood.

Additionally, the likelihoods of a test sample vector \mathbf{x} belonging to each class ($p(\mathbf{x}|\lambda_1)$, $p(\mathbf{x}|\lambda_2)$ for a binary classification problem), can be converted to a posterior probability using **Equation 3.30**. The prior probabilities $P(class1)$ and $P(class2)$ can be determined from the training dataset or be set to obtain the desired trade-off between predictions for class 1 and class 2 [47]. This method can be extended for multi class classification.

$$P(class1|X) = \frac{p(\mathbf{x}|\lambda_1)P(class1)}{p(\mathbf{x}|\lambda_1)P(class1) + p(\mathbf{x}|\lambda_2)P(class2)} \quad (3.30)$$

3.4 Evaluation Methods

In order to determine the efficiency of a model, the model has to be evaluated. Depending on the nature of the data and the type of machine learning models used, different evaluation metrics and methods are applicable. In this section, the different methods and metrics used for evaluating the automatic segmentation system and the cough classifier are discussed.

3.4.1 Metrics

When performing binary classification, the following basic metrics are frequently computed to evaluate the performance of a model:

3.4 Evaluation Methods

TP (True Positives): The number of positive samples classified as positive

TN (True Negatives): The number of negative samples classified as negative

FP (False Positives): The number of positive samples classified as negative

FN (False Negatives): The number of negative samples classified as positive.

These metrics are usually compiled into a confusion matrix as shown in **Table 3.1**:

		Ground Truth	
		Positive	Negative
Classifier Prediction	Positive	TP	FN
	Negative	FP	TN

Table 3.1: An example of a confusion matrix.

These four basic values can be used to calculate a variety of more descriptive metrics. The most commonly computed metric is accuracy, which is computed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.31)$$

Hence the accuracy is the sum of all the correct predictions, divided by the number of predictions made. Accuracy is a very intuitive metric, but in most problems, evaluating the performance of a model using accuracy is not enough. For instance, the accuracy of a model can be a misleading metric if the dataset is unbalanced.

As an example, consider the case where we want to evaluate a model that acts as a spam filter for emails. If we evaluate the model using 1,000 spam emails and 10,000 non-spam emails, and the model predicts that all 11,000 emails are not-spam, the accuracy is: $ACC = \frac{10,000+0}{11,000} = 90,90\%$. This is a misleading figure.

Therefore, when performing model evaluation, we computed the following additional metrics:

$$Sensitivity = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{TP}{TP + FN} \quad (3.32a)$$

$$Specificity = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} = \frac{TN}{TN + FP} \quad (3.32b)$$

In our application, sensitivity would represent the proportion of sick people correctly diagnosed as being sick and specificity would represent the proportion of healthy people correctly diagnosed as being healthy. For the spam filter example, the sensitivity and specificity would be $sens = \frac{0}{0+0} = 0^1$ and $spec = \frac{10,000}{10,000+1,000} = 90.90\%$.

¹In practice, when $TP=0$, the sensitivity is set to 0 to avoid the division by 0.

Another pair of metrics that is frequently used in machine learning model evaluation is precision and recall. Recall is calculated using the same formula as sensitivity and precision can be calculated using $Precision = \frac{TP}{TP+FP}$. Considering the nature of this study, sensitivity and specificity are a good choice for model evaluation. They have also been used by other researchers in studies relating to classification of human disease [38][2][33][48][27][7][30].

3.4.2 Cohen's Kappa Coefficient

Cohen's kappa coefficient (κ) is used as a measurement of agreement between two raters (or judges) that classify N items into K mutually exclusive classes, by taking into account the agreement which can be expected by chance. In the context of binary classification, κ is a single, supplementary metric that compares a reported accuracy with the accuracy that is expected if the classifier makes unbiased random decisions, given the distribution of data between classes.

The formula for calculating κ is given as:

$$\kappa = \frac{(\text{reported accuracy} - \text{expected accuracy})}{(1 - \text{expected accuracy})} \quad (3.33)$$

Consider the email spam filter example mentioned above, but in this case, the classifier predicted the values as shown in the confusion matrix given in **Table 3.2**.

		Ground Truth	
		Spam	Not-Spam
Prediction Classifier	Spam	250	1,000
	Not-Spam	750	9,000

Table 3.2: Spam Filter Confusion Matrix

The reported accuracy of this classifier is: $acc = \frac{250+9,000}{250+1,000+750+9,000} = 84,09\%$. In order to compute κ , we also need to calculate the expected accuracy. The expected accuracy uses the number of instances in each class (in the test set), combined with the number of correct predictions made by the classifier.

Let A denote the event of an email being spam and \bar{A} an email being not spam, let B denote the event of an email predicted as spam and \bar{B} when an email is predicted as not spam. The expected accuracy can then be calculated as follows, with N as the number of samples being classified:

$$\text{Expected Acc} = \frac{1}{N} * \left(\frac{A * B}{N} + \frac{\bar{A} * \bar{B}}{N} \right) \quad (3.34)$$

The value of the expected accuracy for the spam example is thus calculated as:

$$\begin{aligned} \text{Expected Acc} &= \frac{1}{11,000} * \left(\frac{(250 + 750)(250 + 1000)}{11,000} + \frac{(1,000 + 9,000)(750 + 9,000)}{11,000} \right) \\ &= 0.8161 \end{aligned}$$

This means the accuracy that is expected from a classifier making random predictions is 81,61%.

The Cohen's kappa coefficient can subsequently be calculated, using **Equation (3.33)** as:

$$\kappa = \frac{0.8409 - 0.8161}{1 - 0.8161} = 0.1349$$

Interpreting this results, we can say that our reported accuracy is higher than the expected accuracy by 13.49% of the gap between the expected accuracy and 100%. A κ of 0 indicates that the classifier is fairing no better than chance, and a $\kappa < 0$ indicates the classifier is fairing worse than chance.

3.4.3 Receiver Operator Characteristic Curve Analysis

The receiver operator characteristic (ROC) is used as a method for evaluating a two-class classifier in terms of two metrics - true positive rate (TPR) and false positive rate (FPR). TPR is equal to sensitivity and FPR is equal to 1 - specificity [49]. It is widely used in the medical field for evaluation of diagnostic systems and has been adopted in the machine learning community as an evaluation metric to be used for unbalanced class distributions.

In ROC analysis the sensitivity is plotted against the FPR for decision thresholds ranging between 0 and 1. The decision threshold can be defined in various ways, but for this study it is chosen as the probability $P(Y = 1|X)$ at which a positive decision is made, and below this threshold which a negative decision is made.

Thus, to plot the ROC curve, the sensitivity and specificity must be computed for multiple decision thresholds.

Figure 3.7 shows a hypothetical ROC curve. Curve **A** represents an ideal system with a sensitivity and a specificity of 1.0; curve **C** represents a system making random choices with equal sensitivity and specificity while **B** shows a typical ROC curve. The closer the ROC curve is to the upper left corner, the better the system. If an ROC curve is close to curve C, the system performance tends towards that of a random classifier.

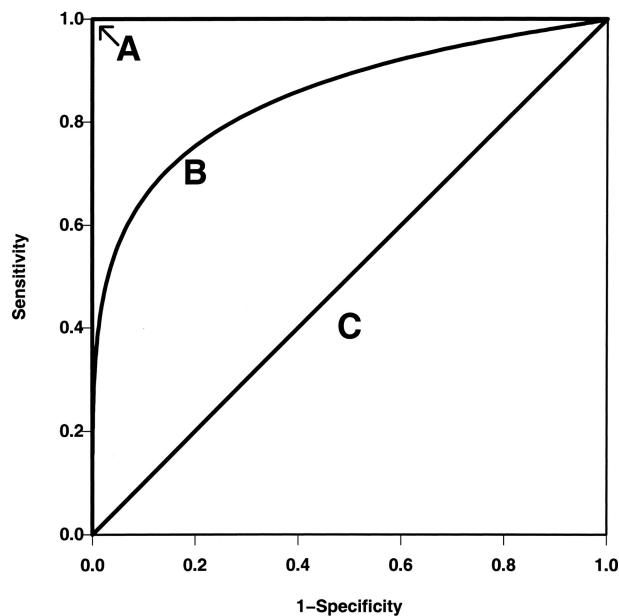


Figure 3.7: Hypothetical ROC curve [50].

By analysing the ROC curve, we can get an indication of how well the system performs. However, during model evaluation, it is desirable to have a single metric to describe the model performance. The area under the ROC curve gives us such a metric.

Formally stated, the AUC is the probability that a classifier will rank a randomly chosen positive sample above a randomly chosen negative sample [51]. Thus the AUC value represents how well the models performs over all possible threshold values, as opposed to accuracy, which is the model evaluated at one specific threshold value. For the hypothetical ROC curve in **Figure 3.7**, the AUC of curve A would be equal to 1 (a perfect classifier), and the AUC of curve C would be 0.5, while the AUC of curve B would be approximately 0.8.

A very attractive property of ROC curves and AUC is that they are insensitive to an unbalanced data set. This is because ROC curves depend on the true positive *rate* and false positive *rate*, which are both ratios that are independent of class distributions. The AUC of a classifier making random choices is thus always 0.5, even when the classes are unbalanced.

For diagnostic models, FPR and TPR are not always equally important. For instance, it might be more dangerous to falsely diagnose a patient as healthy and send them home without medication, than to falsely diagnose a patient as ill and take them into medical care.

Thus, the AUC of a model is a more robust metric when evaluating machine learning algorithms for diagnostic purposes, indicating the model performance considering all decision threshold values and insensitive to the class distribution of the dataset.

3.4.4 Cross Validation

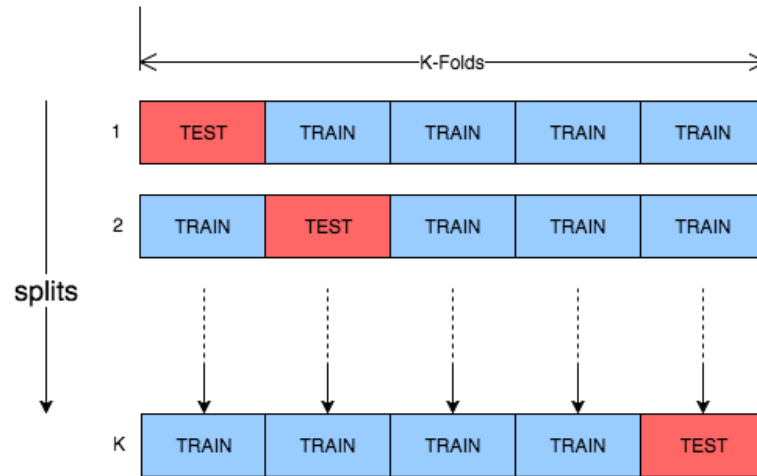


Figure 3.8: Graphical representation of K-fold validation.

In traditional model evaluation, also known as the holdout method, a dataset is split into a two mutually exclusive subsets: one for training and one for testing. The training set is used to estimate the model parameters, usually a weight matrix, through some optimization technique, and consists of around 60% – 90% of the total dataset. The test set, consisting of the remaining 40% – 10%, is then used to evaluate the model by making the model perform predictions on the test set and subsequently calculating the applicable evaluation metrics from the prediction results.

Unfortunately, this method is known to encounter the problem of overfitting, because the test set is usually a randomly selected subset. Optimizing the model hyper parameters on a development set (selected from the test set) can avoid the problem of overfitting, but reduces the size of the test set. Cross validation aims to reduce the effect of overfitting by utilizing the entire dataset in a structured manner.

Figure 3.8 illustrates the process of K-fold cross validation. The process starts by dividing the dataset into K equal parts or 'folds'. The test set is selected as one of these K folds and the training set is selected as the remaining $K - 1$ folds. This is repeated until each of the K folds has been used for testing. Subsequently, the final evaluation metrics are computed by averaging over all K iterations.

A disadvantage of K-fold validation is that it increases the computation time by a factor of K over holdout method. The advantage of this method is however that each sample in the dataset is used for testing exactly once, and thus the results are the best possible representation of what the model's behaviour would be to new data. K-fold validation is particularly useful when performing experiments on a small dataset.

Leave-one-out cross validation (LOOV) is a special case of K-fold validation where K

3.4 Evaluation Methods

is chosen as the number of samples in the dataset. LOOV is considered an exhaustive validation method as it requires N train/test runs for a dataset with N samples. As datasets become large, LOOV becomes less viable.

Leave- p -out validation (LPOV) is a variation of LOOV, where p samples are left out for testing in each iteration. Aside from obvious computation benefits, LPOV is sometimes preferred over LOOV, for instance if a dataset consists of p samples that are not independent. The dataset used in this study is an example of this: for each recording we have multiple coughs that are not independent, because they are produced by the same patient. However, each recording is independent from other recordings, because each is from a different patient. If each cough represents one sample, then leaving out all the coughs from one patient for testing would be the correct way of dividing up the dataset.

Parameter selection

Some models have hyper-parameters, which are structural parameters that can strongly influence the model performance. For example, when designing an HMM system, the number of states and number of mixtures used in each state would be considered hyper-parameters.

In order to achieve the best performance, the hyper-parameters of a model need to be optimized. One method for selecting the optimal hyper-parameters is called grid search. Grid search is a brute force method that trains and tests the model for all possible combinations of the hyper-parameter space. **Figure 3.9** shows the parameter selection process in a flow diagram.

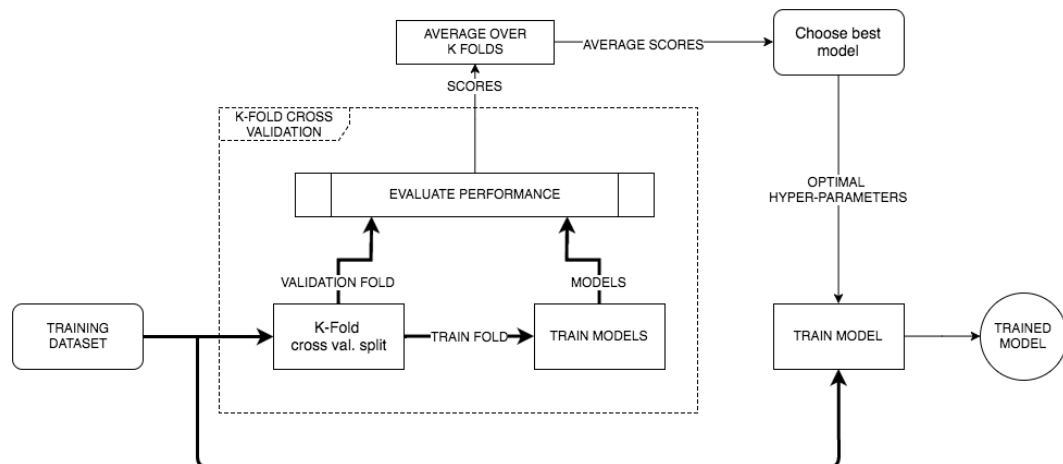


Figure 3.9: Parameter selection flow diagram.

Parameter selection is usually performed on a subset of the training set, referred to in this study as the validation set. The validation set is selected in the same way the test set is selected and would also usually comprise of about 10% – 30% of the training set. The remaining 90% – 70% is used to train models over a range of hyper-parameters.

This is repeated for each of the K folds, and the best performing model over all K folds is selected. A common mistake is to perform parameter selection on the held out test set, which results in overfitting.

To overcome the issues mentioned in **Section 3.4.4**, K -fold validation is often used during parameter selection. Thus, for each possible combination of hyper-parameters, K -fold validation is performed to compute the model performance. Then the parameters of the model with the best performance is selected and training is done on the entire training set.

3.5 Summary

In this chapter the methods used for the design of an automatic cough annotation system as well as a cough classification system were discussed.

HMMs can be used to segment continuous, time series data into several classes by modelling the state transitions as Gaussian distributions.

Logistic regression is a classification method that uses the logistic sigmoid function to estimate the probability that a sample belongs to a certain class.

Gaussian mixture models model the training data by fitting a linear combination of multivariate Gaussian distributions to the data. Classification is performed by computing the probability that a new sample belongs to one of the modelled classes and selecting the class with the highest probability.

The methods and metrics that will be used to evaluate our models were discussed. Sensitivity and specificity were introduced as additional metrics to accuracy, and the Cohen's Kappa coefficient was proposed as a measure of model performance while using an unbalanced dataset.

ROC curve analysis was discussed as a method to optimize the trade-off between sensitivity and specificity by adjusting the threshold parameter when binarizing a model's output probabilities. The AUC is introduced as a metric that is insensitive to class imbalance and more applicable when evaluating diagnostic systems. In this study, AUC is used as the major evaluation metric.

Cross validation is introduced as a method to utilize the entire dataset for training and testing. All the methods in this study are evaluated through K -fold validation and the average of each metric is computed over all folds. Lastly, the grid search parameter selection method is discussed. Grid search is used to select the optimal hyper-parameters of a machine learning model by iteratively training and testing over all possible combinations and selecting the model with the best performance.

Chapter 4

Data Acquisition

For this study, a data corpus consisting of coughing sounds was recorded in a controlled environment. To achieve this, we collaborated with a research team from the University of Cape Town (UCT) who granted us access to this facility and the opportunity to record patients at the Brooklyn Chest Health Clinic in Cape Town, South Africa.

The study conducted by the research team at UCT aims to investigate the use of cough aerosol sampling (CASS), a novel technology used to analyse the individual cough aerosol particles, to better understand the pathobiology of transmission in patients with drug resistant TB.

In this study there are two patient groups: TB positive (sick) and TB negative (healthy). The TB positive group consists of patients with suspected TB, who were included into the study for cough aerosol sampling. The TB negative group consists mostly of patients who were included because they had contact (household or close/personal) with the patients undergoing the CASS procedure. These patients are completely healthy with no known pulmonary illnesses.

The clinical diagnosis (ground truth) was established using sputum culturing as this is the standard for active TB diagnosis. Some patients initially in the TB positive group were later diagnosed to be TB negative and were subsequently added to the TB negative group.

4.1 Data Acquisition

All audio recordings were collected specifically for this project and were conducted by Sister Wilson at Brooklyn Chest Health Clinic. The recording environment is described in **Section 4.1.1**. The audio dataset used to develop the automatic annotation system and the automatic cough classifiers are described in **Section 4.1.2**. Additionally, a



Figure 4.1: Recording setup at the Brooklyn Chest Health Clinic.

clinical dataset (meta data) was collected by the research team at UCT and is described in **Section 4.1.3**.

4.1.1 Recording Setup

The recordings took place inside the cubicle shown in **Figure 4.1**, which was constructed by the UCT research team for performing cough aerosol sampling. This procedure would typically consist of a patient sitting in the cubicle for 4 - 6 minutes, while coughing into a plastic pipe, which is attached to a machine collecting airborne sputum droplets. The machine collecting the sputum droplets was situated in a room outside the cubicle but nevertheless created an audible background noise.

Additionally to the CASS session, patients completed a voluntary cough procedure, during which they would perform the same procedure as in the CASS session but without having a pipe in their mouth and with the machine switched off. This was done in order for the patients to produce sputum samples for subsequent analysis. Both these sessions were recorded by our recording equipment.

For the recording of cough sounds, a Tascam DR-44WL hand held audio recorder and a Rhode M3 microphone were used. Recording was performed at a sampling rate of 44.1kHz at a resolution of 16 bits per sample.

The recording unit was mounted outside the cubicle, so that the recordings could be started, stopped and paused without exposing Sister Wilson to TB bacteria. The microphone was wall mounted inside the cubicle at a height of 95cm (approx). The hand held recorder was used to control the microphone only and did not record any sounds outside the cubicle. Additionally, a shock absorbing microphone brace was installed to limit unwanted noise in the form of vibration through the dry wall on which the microphone was mounted. The cubicle is shown in **Figure 4.1** with measurements: 100 breadth x 116 width x 220 height (cm).

4.1.2 Audio Dataset

Our complete audio dataset consists of 81 recordings from 60 different patients. This dataset includes recordings from the CASS sessions as well as the voluntary cough sessions. For the contact patients, which makes up the bulk of the TB negative group, only voluntary cough sessions were recorded.

In the following, two audio datasets \mathcal{D}_{CC} and \mathcal{D}_{AA} are discussed. \mathcal{D}_{CC} refers to the dataset used to train the cough classifiers, described in **Section 6.4** and \mathcal{D}_{AA} is the dataset used to train the automatic cough segmentation system, described in **Section 5.2.2**. Summaries of all the datasets discussed here are given in **Section 4.2**

Cough Classifier Dataset \mathcal{D}_{CC} : Unfortunately, the number of TB negative patients who performed the CASS procedure was very small, as this only consisted of patients who were initially thought to have TB, but were later diagnosed as TB negative. Thus, it was decided not to use any of the CASS recording sessions to train the automatic cough classifiers. The decision was taken after initial tests showed that an automatic cough classifier trained on all the recording data was predominantly performing classification based on the background noise of the CASS machine and the audio filtering caused by coughing into a pipe.

The remaining data after removing all the CASS session recordings consisted of 38 recordings, of which 17 were from patients diagnosed with TB and 21 from patients confirmed to not have TB.

From **Table 4.2**, we can see that we now have more recordings from TB negative patients than from TB positive patients. This is because we have removed the CASS recordings, which were all from TB positive patients. However, the total length of audio obtained from TB positive patients is still more than for TB negative patients. This shows that the duration of the additional voluntary cough sessions were usually much longer for TB positive patients than for TB negative patients.

Automatic Annotation Dataset \mathcal{D}_{AA} : During the automatic annotation process, cough classification is not performed¹. For this reason, audio from both the CASS and voluntary sessions could be used to develop the automatic annotator. Since an automatic cough annotator should function under various recording environments, the inclusion of recordings with background noise and audio filtered by a pipe would result in a more robust annotation system.

This dataset consists of a subset of recordings from the complete dataset that were manually annotated. The full audio dataset was not used as this would have taken too long to annotate manually. The recordings were chosen in a chronological order. As new recordings were added to the complete dataset, they were manually annotated and added to this dataset, until we developed a working baseline prototype system. Thereafter, the dataset used for developing the automatic annotation system was not altered, in order to compare different methods objectively.

A summary of this dataset is shown in **Table 4.3**. We see that, as for the \mathcal{D}_{CC} dataset, there is a very small amount of audio data for TB negative patients, compared to the TB positive data. Due to this imbalance, we did not design the annotation system to perform classification, although this is possible with an HMM system.

4.1.3 Clinical Dataset

According to the World Health Organization (WHO), a chronic cough, coughing up blood, chest pains, weakness, weight loss, fever and night sweats are the most prevalent symptoms in TB positive patients [52]. However, these symptoms are largely self reported and are thus not objective.

Objective measurements are important in order to diagnose and track the recovery of TB patients. In a recent study researchers defined a new indicator called the "TB Score" [53]. By associating a weight with each symptom and representing each symptom as a binary value (0 for no, 1 for yes and using threshold values for measured quantities), these values are summed to compute a score out of 13. The TB score was computed at set intervals over the treatment duration and used as a clinical index to predict mortality of test subjects.

The TB score was calculated using the same symptoms as proposed by the WHO in [52], although the researchers in [53] replaced some of the self reported symptoms with objective measurements. Weight loss and fever were replaced by body mass index (BMI). Axillary temperature and mid upper arm circumference (MUAC) were added as additional

¹The automatic cough annotation system could be extended to include classification, but this was not done in this study due to data constraints.

4.1 Data Acquisition

clinical measurement. MUAC is measured with a non-stretchable band at the bi-cep of the non-dominant arm. Anaemic conjunctivae (pale coloured eye lids) was also included as an objective observation and used in the calculation of the TB Score.

Our Clinical Dataset: In addition to the audio recordings of coughing, extensive meta-data was collected for each patient included in the study. The clinical dataset consists of 518 samples, of which 122 (23.55%) are from TB negative and 396 (76.45%) from TB positive patients. The meta data was collected over a longer period than the audio data and thus includes a larger number of patients.

This meta-data was made available to us by the research team at UCT and included information such as:

- TB Symptoms (Yes/No): current cough, cough duration > 2 weeks, coughing blood, weight loss, loss of appetite, night sweats, shortness of breath, anaemic conjunctivae, positive finding at lung auscultation, fever.
- TB Symptoms (measured): temperature, respiratory rate, heart rate, height, weight, MUAC.
- If TB positive: TB type, currently on TB medication?

Following the reasoning by the researchers in [53], we did not use self reported symptoms but retained only objective measurements and observations for use in our experiments. Thus anaemic conjunctivae, heart rate, temperature, BMI and MUAC were used to develop our clinical data based classifier.

4.2 Database Summaries

The databases described in this chapter are summarized in the following tables. The complete dataset (**Table 4.1**) describes the extent of all recorded data, while the Cough Classifier and Automatic Annotator datasets in **Tables 4.2** and **4.3** describe the subsets of the complete dataset used for experimentation.

Table 4.1: Complete Audio Dataset

	TB Positive	TB Negative	Total
Recordings	55	26	81
Patients	35	25	60
Total Recording time	3h:8min:20sec	0h:32min:20sec	3h:40min:40sec
Mean Recording time	0h:3min:25sec	0h:1min:14sec	0h:2min:43sec
Std Deviation Recording time	0h:1min:32sec	0h:1min:14sec	0h:1min:46sec
Max Recording time	0h:8min:32sec	0h:4min:36sec	0h:8min:32sec
Min Recording time	0h:0min:29sec	0h:0min:8sec	0h:0min:8sec

Table 4.2: Cough Classifier Dataset \mathcal{D}_{CC}

	TB Positive	TB Negative	Total
Recordings	17	21	38
Patients	17	21	38
Total Recording time	0h:52min:24sec	0h:15min:40sec	1h:8min:4sec
Mean Recording time	0h:3min:4sec	0h:0min:44sec	2min10sec
Std Deviation Recording time	0h:2min:21sec	0h:0min:24sec	0h:1min:59sec
Max Recording time	0h:8min:32sec	0h:1min:25sec	0h:8min:32sec
Min Recording time	0h:0min:29sec	0h:0min:8sec	0h:0min:8sec

Table 4.3: Automatic Annotator Dataset \mathcal{D}_{AA}

	TB Positive	TB Negative	Total
Recordings	19	22	41
Patients	16	22	38
Total Recording time	1h:1min:58sec	0h:24min:26sec	1h:26min:24sec
Mean Recording time	0h:3min:15sec	0h:1min:6sec	0h:2min:6sec
Std Deviation Recording time	0h:1min:42sec	0h:1min:8sec	0h:1min:47sec
Max Recording time	0h:8min:24sec	0h:4min:36sec	0h:8min:24sec
Min Recording time	0h:0min:29sec	0h:0min:8sec	0h:0min:8sec

Table 4.4: Clinical Data Information

Name	Description	Measured Units
MUAC	Mid upper arm circumference, measured with a measuring tape.	Millimetre
Temperature	Temperature measured with a thermostat.	Celsius
BMI	Body mass index: $height/weight^2$	kg/m^2
Anaemic conjunctivae	Paleness of the conjunctiva determined through eye examination.	Binary (yes/no)
Heart rate	Rate of the patient's heartbeat.	$beats/minute$

4.3 Summary

In this chapter, the collection and pre processing of our audio and clinical datasets was discussed.

Our cough classifier audio dataset consists of 38 recordings (17 TB positive and 21 TB negative), providing a total of 746 coughs. This compares favourably with the database used in a study considering the analysis of coughing sounds made by pneumonia patients in [2], but could be improved in future.

Our automatic annotation dataset consists of 41 recordings from 46 different patients, with a total length of 1 hour, 46 minutes and 44 seconds. Additionally our clinical dataset, summarized in **Table 4.4**, consists of 518 samples (396 TB positive and 122 TB negative) and has been reduced to only include objective features.

Chapter 5

Data Preparation

This section describes the data pre-processing and segmentation steps. During pre-processing, the raw data is cleaned to remove unwanted noises (such as doors banging) and then the data is normalized. After normalization, the audio data is segmented in order to extract cough sounds. Manual annotation was performed to create the dataset with which the automatic cough classifier was trained.

5.1 Data Pre-processing

This section describes the signal processing steps applied to the raw audio recordings that prepare the data for experimentation and are common to all tested systems.

5.1.1 Audio Cleaning

Prolonged silences and irrelevant audio were removed from all recordings in **Table 4.3**. These periods typically occurred at the start and end of the recording or when the nurse opened and closed the door of the cubicle. These noises are not expected to be encountered when using the system in practice, thus were removed as a first step. Removing these sections also speeded up the annotation process, as these unwanted sounds are visibly distinguishable in the waveform and thus reduces the time spent listening to each recording while segmenting. **Figure 5.1** shows an example of how a raw recording is manually trimmed to contain only relevant data.

5.1 Data Pre-processing

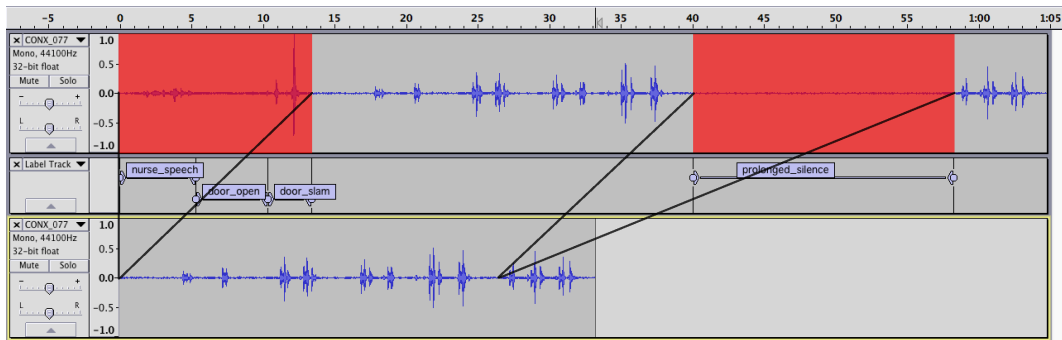


Figure 5.1: Waveform before and after manual trimming. The red segments indicate sections that will be removed from the top waveform, resulting in the bottom waveform.

5.1.2 Normalization

During the recording process, differences in waveform amplitudes were observed. This was due, for example, to patients sitting at different distances from the microphone or the naturally differing loudness of the patient's cough. In order to compensate for this difference, each recording was normalized. The normalization algorithm can be summarized as follows:

- Divide audio into non-overlapping frames of length 512 samples.
- Determine if a frame is an *event* or *silence* by comparing the energy of the frame with a threshold value.

Event-threshold = $2 \times$ standard deviation of the energy of the entire signal.

- Compute (σ_e, μ_e) and (σ_s, μ_s) , the standard deviation and mean of the event- and silence energy respectively.
- Estimate event and silence Probability Density Functions (PDFs) from (σ, μ) (assume Gaussian) .
- Obtain a new threshold value by computing the intersection of the silence- and event-energy PDFs.
- Recursively re-threshold the frame-energies using the new threshold value. Update the threshold to the intersection of the new energy and silence PDFs after each iteration.
- Stop when the change in the number of event and silence frames falls below 10. The number 10 was chosen empirically.
- Then multiply all samples with the ratio $B = \frac{A}{S}$, where:

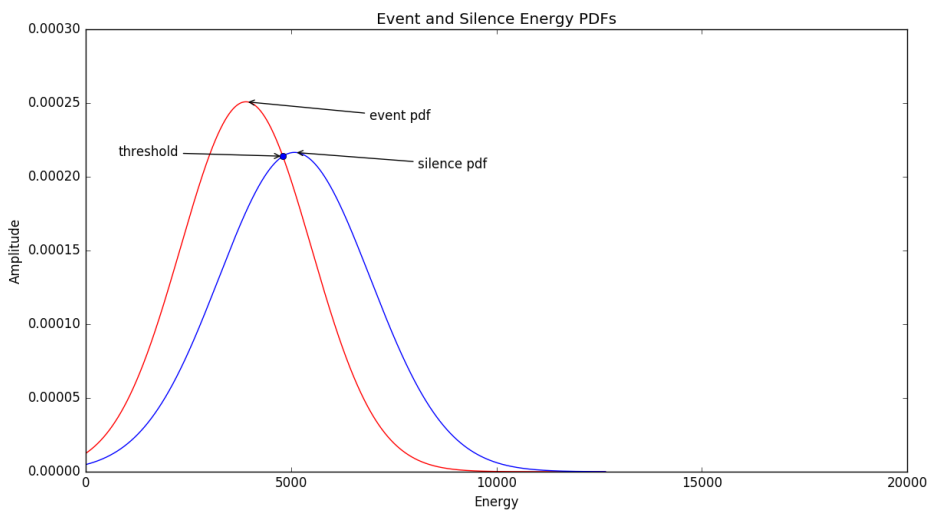
A is the maximum standard deviation of all event energy frames, and

5.1 Data Pre-processing

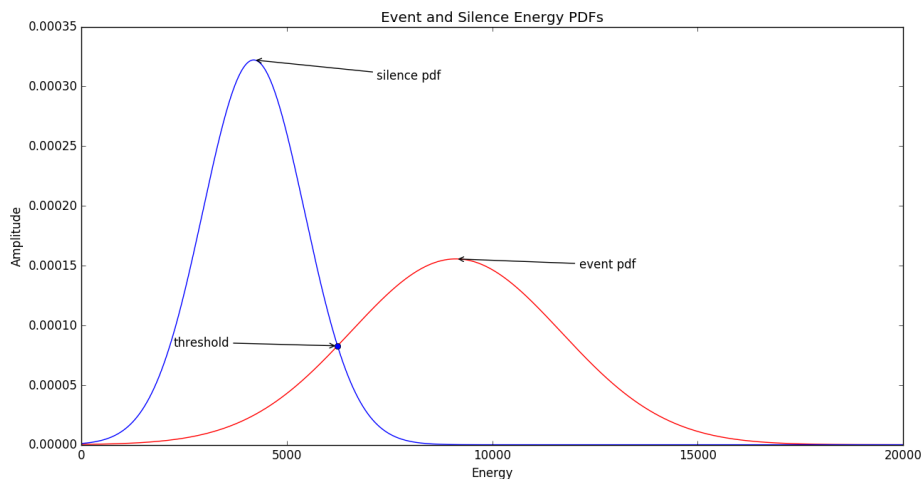
S is the energy standard deviation of each frame.

- Thus the waveform has been scaled so as to normalize the energy of the events among recordings.

Figures 5.2(a) and **5.2(b)** show an example of the event and silence PDFs, before and after the normalization algorithm. The threshold is also shown as the intersection of the two PDFs. From **Figure 5.2(b)** we can see that after iteratively separating event and silence frames, the silence energy is centered around a smaller, lower energy value than the event energies, which are distributed at higher values with a larger standard deviation.



(a) Energy PDFs after initial thresholding with $2 \times \text{std-dev}$ of signal energy.



(b) Energy PDFs after iterative normalization is complete.

5.2 Annotation and Segmentation

This section describes how the continuous audio datasets were manually annotated and how an automatic annotation system was developed to determine information about the datasets \mathcal{D}_{CC} and \mathcal{D}_{AA} regarding individual events. Annotations refer to text labels and associated start and end times. Segmentation refers to the process of isolating certain events (such as coughs) in time from the continuous audio.

The continuous audio datasets \mathcal{D}_{CC} and \mathcal{D}_{AA} were manually annotated using the audio editing software tool Audacity. The manual annotations of \mathcal{D}_{AA} were used as the ground truths for developing the automatic annotation system, and the manual annotations of \mathcal{D}_{CC} were used as ground truths for developing the cough audio classifiers.

In practice, the automatic annotation system would be used to perform annotation of continuous audio for subsequent extraction of cough events and classification. However, in this study, manual annotations were used as ground truth for the cough classifier as this was more reliable.

5.2.1 Manual Annotation

The manual annotation process involved carefully listening to each individual audio sample, and identifying which audio events occurred at which times. For \mathcal{D}_{CC} , only cough events were annotated. However, when annotating \mathcal{D}_{AA} , sounds that commonly occurred in our recordings were also labelled. This was done in order to build a more robust annotation system for future use in the project. For instance, if a patient is recorded while coughing and clears his/her throat, it would be beneficial if the system can identify the difference between a cough and a throat-clear. This functionality would be necessary for the system to function properly in a real world scenario.

Thus, for \mathcal{D}_{AA} , all recordings were annotated with the following classes/events: **cough**, **silence**, **ambient sounds** (chair moving, opening/closing sputum container etc), **speech**, **throat clearing** (includes all other sounds similar to throat clearing eg snorting, grumbling) and **spitting**.

More information about the manual annotation process and how Audacity was used can be found in [Appendix A](#).

After manually annotating \mathcal{D}_{AA} , it was discovered that there were insufficient data samples to train an HMM for each of the following classes: **ambient sounds**, **speech**, **throat clearing**, **spitting**. These classes were therefore combined to form a new class **other**. Thus, the automatic annotation system used only three classes - **cough**, **silence** and **other**. Because the primary goal of this system is to extract cough sounds, we are not sacrificing any information at this stage of the study. The more detailed annotation

5.2 Annotation and Segmentation

nevertheless allow future refinements of the system.

Tables 5.1 and **5.2** summarise \mathcal{D}_{CC} and \mathcal{D}_{AA} after manual annotation respectively. We can also see that the average coughs per recording for TB negative patients is 11, while this figure for TB positive patients is 29.

Table 5.1: Segmented \mathcal{D}_{CC} Corpus

	TB Positive	TB Negative	Total
Recordings	17	21	38
Patients	17	21	38
Coughs	501	245	746
Avg coughs/patient	29	11	19
Total cough Time	4min:38sec:36ms	1min:39sec:300ms	6min:17sec:337ms
Mean cough Duration	0min:0sec:554ms	0min:0sec:405ms	0min:0sec:505ms
Cough duration Std-dev	0min:0sec:219ms	0min:0sec:153ms	0min:0sec:212ms
Max cough duration	0min:1sec:420ms	0min:0sec:980ms	0min:1sec:420ms
Min cough duration	0min:0sec:180ms	0min:0sec:100ms	0min:0sec:100ms

Table 5.2: Segmented \mathcal{D}_{AA} Corpus

	TB Positive	TB Negative	Total
Recordings	19	22	41
Patients	16	22	38
Total cough time	0h:10min:58sec:538ms	0h:3min:59sec:582ms	0h:14min:58sec:121ms
Total silence time	0h:49min:55sec:181ms	0h:20min:0sec:423ms	1h:9min:55sec:604ms
Total 'other' time	0h:1min:16sec:533ms	0h:0min:41sec:958ms	0h:1min:58sec:491ms

5.2.2 Automatic Segmentation

Manual annotation is very time consuming. To accelerate this process, the recordings in \mathcal{D}_{AA} were used to train hidden Markov models (HMMs) with which newly recorded data could be automatically annotated. The annotations produced by the HMMs can then be used to extract the cough audio events, which are the expected input for subsequent cough classifiers.

Note that in this study, the automatic annotation system described here was not used to annotate the data for \mathcal{D}_{CC} and was implemented as a proof of concept for future work. As mentioned previously, \mathcal{D}_{CC} was annotated manually to provide the best results.

The design and implementation of the HMM system can be split into three main steps: constructing the training corpus, defining and training the models and model evaluation. For the design and implementation of the HMM cough annotator we used the hidden Markov model Toolkit (HTK), developed by Cambridge University Engineering Department [54].

Feature Extraction

During feature extraction, each recording in the dataset \mathcal{D}_{AA} was split into overlapping frames. The frame length was chosen as 25ms with an overlap of 10ms between frames. At a sampling rate of 44.1kHz, this translates to a frame length of 1102 samples and overlap of 441 samples. A Hamming window of length 1102 samples was applied to each window to reduce truncation effects when applying the Fast Fourier Transform (FFT). For each window, 13 Mel-Frequency Cepstral Coefficients (MFCCs) were computed. MFCCs are short-time spectral features that model the human hearing spectrum by mimicking our almost log-scale perception of loudness and pitch and excluding the fundamental frequencies that are related to speaker dependent characteristics. More information about MFCCs and how they are computed can be found in [Section 6.2](#).

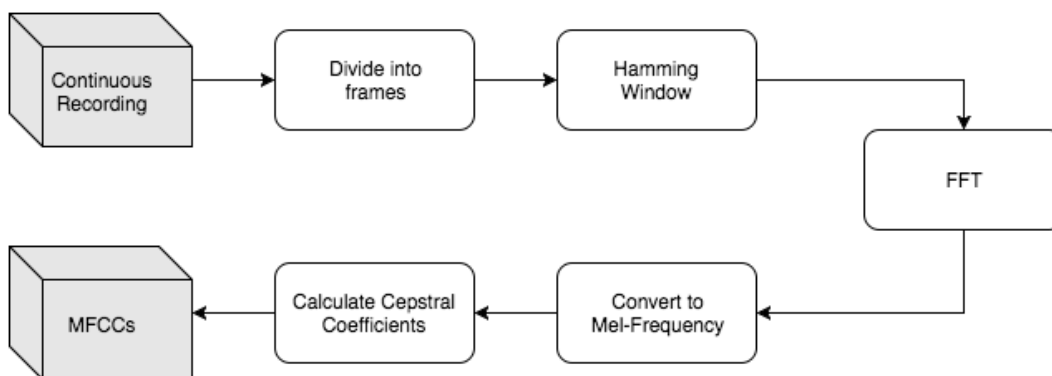


Figure 5.2: Computation of mel frequency cepstral coefficients (MFCCs).

To calculate the MFCCs, the block diagram in [Figure 5.2](#) was followed. We included the first 13 MFCCs as well as the first and second derivatives (deltas and delta-deltas). The first and second derivatives represent how the coefficients change between frames. As we are dealing with a signal that is not stationary, including the derivatives as features would provide us with information about how the signal changes over time.

Thus, for each window, we compute a 39-dimensional feature vector: $\mathbf{f} = [\mathbf{x}, \dot{\mathbf{x}}, \ddot{\mathbf{x}}]$. By doing this for all windows in each recording contained in \mathcal{D}_{AA} , we construct a feature matrix \mathcal{F}_{AA} .

In [Section 3.1.2](#) we showed that for an HMM to optimize the state transition and emission probabilities (a_{ij} and b_{jk}), along with our feature matrix \mathcal{F}_{AA} , we need the state-time alignments corresponding to the data in \mathcal{F}_{AA} , generally referred to as the labels. These labels have already been determined during the manual annotation step. Thus, combining \mathcal{F}_{AA} and the corresponding labels of \mathcal{D}_{AA} , our training corpus is complete.

Defining and training the models

We are interested in detecting three different acoustic events: cough, silence and other. Thus we need to define and train an HMM for each of these events. Before training the HMMs, we need to define the structure of the HMM system and, for each model, we need to define the following hyper parameters:

- Number of states
- Number of Gaussian mixtures used to describe the observation functions for each state and the form of their covariance matrices (diagonal, full, spherical etc.)
- Transition probabilities between states.

Figure 5.2.2 shows a state diagram of our HMM system. There are three classes (cough, silence and other) which are each modelled by a separate HMM. The system will always start and end with silence (indicated separately as START SILENCE and END SILENCE, but in practice these are the instances of the same model). Between the start and end silences, the system may transition between cough, silence and other or stay in any of these models. Intuitively, this means that the HMM system will expect a recording to start and end with silence, and in between it must classify each frame as cough, silence or other, while allowing consecutive frames to belong to the same class.

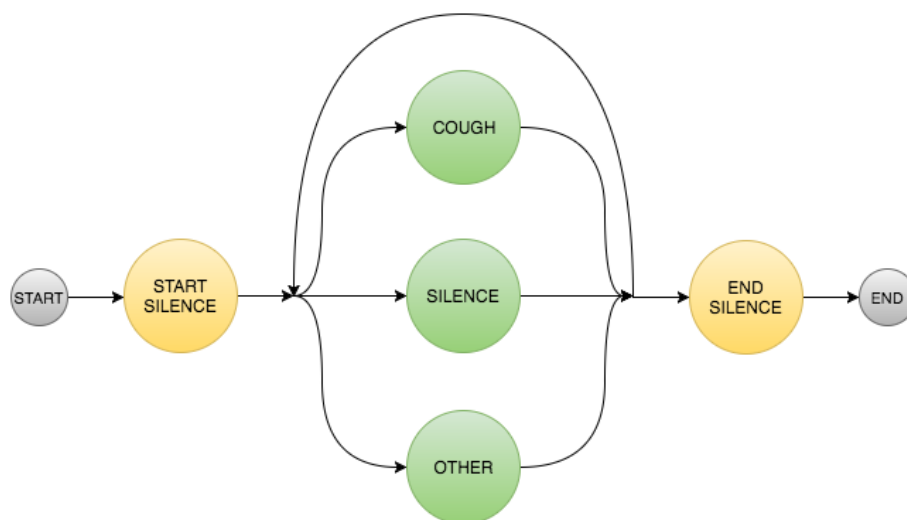


Figure 5.3: HMM State Diagram

Each class is modelled as an HMM with the hyper parameters mentioned above. Choosing the number of states and number of mixtures can be done arbitrarily, but this might result in sub optimal performance. Thus, these hyper parameters were chosen through cross validation and grid search as described in **Section 3.4.4**. Each model can be optimized individually, however, to reduce computation time we chose to use the same hyper parameters for each class.

5.2 Annotation and Segmentation

The transition probabilities define the probability of moving from one state to another. These values are initialized randomly and are optimized by the Baum-Welch algorithm described in **Section 3.1.2**. However, before training we must define which state transitions are allowed. In this system, transitions are only allowed from state i to state $i+1$. Staying in the current state is also allowed, but no backward transitions are allowed. There is no recipe for choosing these rules. This choice is motivated by the assumption that the sounds we are modelling follow a repeatable temporal pattern.

Calculating Annotator Accuracy

To calculate the accuracy of the automatic annotator, the manually annotated files were used as reference. The annotation system outputs a plain text file specifying the start and end times of an event and its label. In speech recognition, the sequence of output labels would be compared to the sequence of input labels to calculate how well the model is performing. However, when using this model to segment cough events, we are also interested in how well the model performs at defining the start and end times of a cough, as we require the entire cough to be segmented from the audio.

Therefore, the sequence of events cannot be used as an evaluation metric. The start and end times of a sequence need to be taken into consideration

Thus, a simple algorithm was developed to convert the (start, end, label) format to frame-based labels. A frame length of 25ms and 10ms frame periodicity was chosen to correspond with the frame lengths used in calculating the MFCCs in **Section 5.2.2**.

If a frame is entirely encapsulated inside an event, the frame is labelled with the label of that event. If a frame is overlapping two events, the event to which the frame is leaning the most was chosen. If a frame is exactly halfway between two annotated events, that frame is omitted from the evaluation.

After constructing the frame-based labels for the reference and test files, these labels were compared and the sum of all correct comparisons, normalised by the total number of comparisons was used to quantify the accuracy of the annotator.

HMM Segmentation Results

The full dataset, \mathcal{D}_{AA} , was split into 3 different subsets: a testing set, a training set and a validation set. The test and training set were selected using 5-fold cross validation. The validation set was selected by randomly selecting 10% of the training set. Thus, each fold contained a test-, train- and validation set and the results reported in **Tables 5.4** and **5.5** are the results after averaging the over all folds for the test and validation sets

5.2 Annotation and Segmentation

Table 5.4: HMM Frame-based Accuracies: Validation set results

No. States	No. Mixtures									
	1	2	3	4	5	6	7	8	9	10
1.0	86.34	69.42	64.63	61.48	60.22	59.87	55.93	51.75	48.61	48.56
2.0	81.76	70.1	57.86	59.1	60.78	61.63	63.72	64.66	65.29	65.27
3.0	84.99	82.94	72.63	67.47	66.56	65.74	64.78	62.85	62.5	62.28
4.0	83.08	71.69	68.9	71.31	70.69	69.33	69.12	68.65	68.33	67.77
5.0	86.02	70.24	67.44	66.23	65.83	66.03	63.6	62.79	62.17	61.73
6.0	87.8	79.05	76.34	76.52	75.44	75.28	75.17	75.13	74.77	75.05
7.0	88.3	77.27	70.82	67.31	64.66	63.22	62.03	59.28	59.05	58.11

Table 5.5: HMM Frame-based Accuracies: Test set results

No. States	No. Mixtures									
	1	2	3	4	5	6	7	8	9	10
1.0	85.12	67.93	63.35	59.62	58.58	58.02	55.05	52.76	49.45	48.08
2.0	83.48	77.0	57.0	53.82	52.44	50.21	53.62	57.73	59.7	61.15
3.0	83.97	83.18	77.55	73.47	73.22	73.23	72.84	72.49	73.29	72.52
4.0	85.59	80.77	78.76	75.09	73.25	72.43	73.43	72.58	74.02	74.16
5.0	87.06	75.23	71.99	68.61	66.8	66.3	65.42	69.77	69.9	69.97
6.0	87.94	80.44	74.75	72.08	72.38	72.1	72.04	72.76	71.94	71.65
7.0	87.16	77.24	71.02	65.95	65.24	66.37	67.65	67.82	67.82	66.64

respectively.

Hyper-parameter selection was done using the grid search with cross validation method. The scope of the model parameters are described in **Table 5.3**. For cross validation, the number of folds was set to 5.

Table 5.3: HMM hyper-parameter scope

Parameter Name	Value Range
No. States	{1..7}
No. Mixtures	{1..10}

The results of the system, evaluated on the validation set is shown in **Table 5.4**. From these results we can see the best performance is acquired using 7 states and 1 mixture with a reported accuracy of 88.3%.

The results of the system on the test set, over the same hyper-parameters, are shown in **Table 5.5**. The accuracy of the model with the selected hyper-parameter is 87.16%. We can observe that the results for the optimal parameters selected using the validation set are actually not the best results reported on the test set (87.94% using 6-states and 1 mixture). However, as discussed earlier, if we were to select the optimal parameters on using the test set, we would be prone to overfitting.

5.3 Summary

In this section we discussed how the audio dataset was prepared for training a cough classifier. First the data was cleaned from all audio events that would not be expected during use, such as doors closing. Then the data was normalized to remove the effects of patients sitting closer/further away from the microphone when coughing.

After cleaning and normalizing the data, HMMs were used to train a system for automatic cough segmentation from continuous audio. Hyper-parameter selection was performed using grid search and cross validation. The optimal hyper-parameters were selected as 7 states and 1 mixture, with a reported accuracy on the held out test set of 87.16%

Chapter 6

Cough Classifier Design and Evaluation

6.1 System Overview

In this chapter, the feature extraction and audio-based classifier design and evaluation are discussed. Feature extraction is performed for all data in the \mathcal{D}_{CC} dataset and is described in **Section 6.2**. Three classifiers are designed and evaluated using nested K-fold cross-validation. The evaluation procedure is described in **Section 6.3** and the results are reported and discussed in **Section 6.4**.

A basic system overview is shown in **Figure 6.1**.

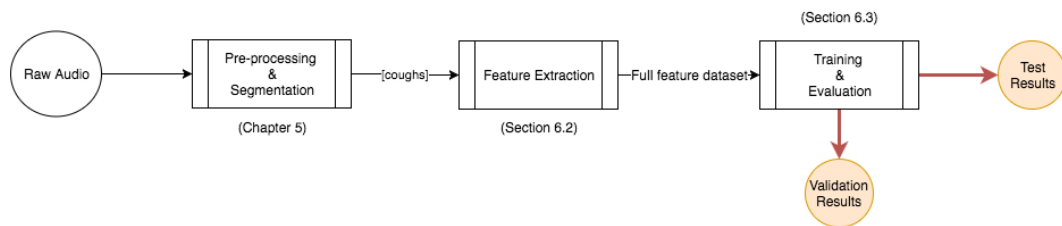


Figure 6.1: Cough classifier evaluation flow diagram.

6.2 Feature Extraction

This section briefly describes the features that were extracted for training the cough classifiers. One common trait among all these feature extraction methods is that each audio signal was divided into non-overlapping frames of N samples. The value of N was later optimized.

6.2.1 Log Filterbanks

To calculate log filterbank features from a signal, we generate F triangular filters, linearly spaced in the frequency spectrum. These triangular filters can then be applied to the magnitude or the power spectrum of the audio by multiplication. **Figure 6.2** shows an example with 5 filters.

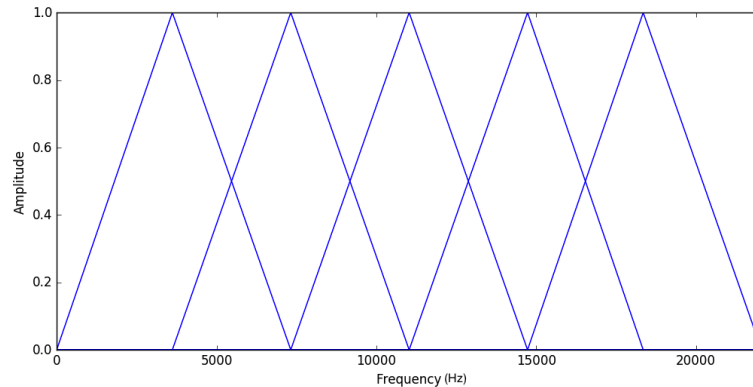


Figure 6.2: Linearly-spaced triangular filterbank.

We chose to apply the filterbank to the log-power spectrum. To calculate the power spectrum of a discrete signal $\mathbf{x}_i(\mathbf{n})$, we first calculate the discrete-time transform (DFT) of $\mathbf{x}_i(\mathbf{n})$, using **Equation 6.1**, where N is the number of samples in \mathbf{x}_i , K is the length of the DFT and $h(n)$ represents a data window function. For our implementation, we chose a Hamming window.

$$X(k) = \sum_{n=1}^N x_i(n)h(n)e^{-j2\pi kn/N} \text{ with } \{1 \geq k \geq K\} \quad (6.1)$$

We can then calculate the power spectrum of $X(k)$ by using **Equation 6.2**.

$$S(k) = \frac{1}{N}|X(k)|^2 \quad (6.2)$$

Figure 6.3 illustrates the power spectrum of a single audio frame, before and after multiplication by a single triangular filter. Note that the spectrum reaches to 22.05kHz, due to a sampling rate of 44.1kHz.

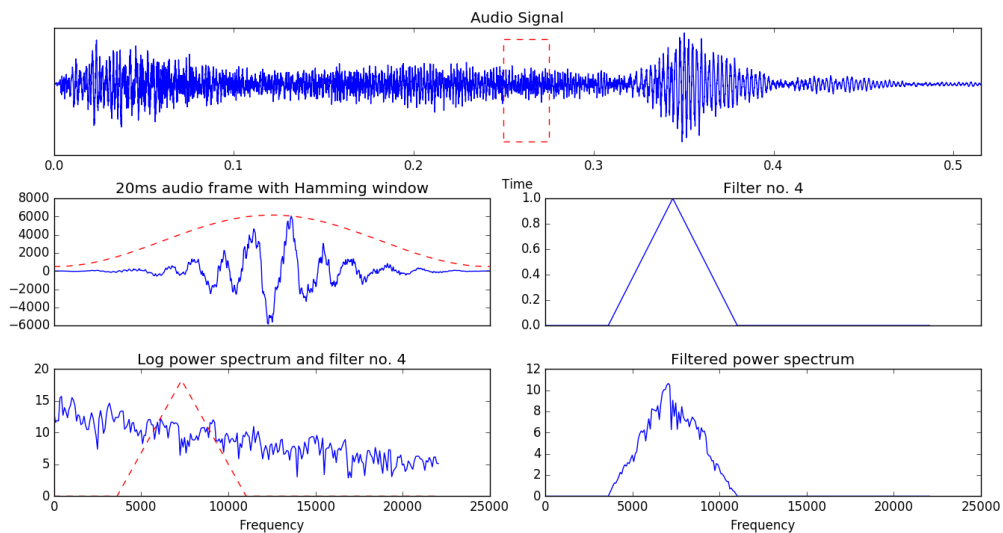


Figure 6.3: Feature extraction by log filterbanks. The area under the curve in the bottom right graph constitutes a single feature.

After application of the triangular filter, the resulting log power values are summed in order to obtain a measure of the total energy in the frequency band spanned by the filter. This is repeated for each of the F filters, resulting in a feature vector of F log energies.

6.2.2 Mel-frequency Cepstral Coefficients (MFCCs)

MFCCs are perceptually-motivated, short time spectral features widely used in speech- and speaker recognition, and were first introduced in [55]. The effectiveness of MFCCs in speech technology lies in their ability to accurately represent the perceptually important parts of the envelope of the short time power spectrum of a signal. In speech signals, the envelope of the short time power spectrum represents the shape of the vocal tract (shape of the mouth, tongue, teeth etc. that create the sound). Thus MFCCs efficiently model the vocal tract information, which is key to recognizing phonemes in speech.

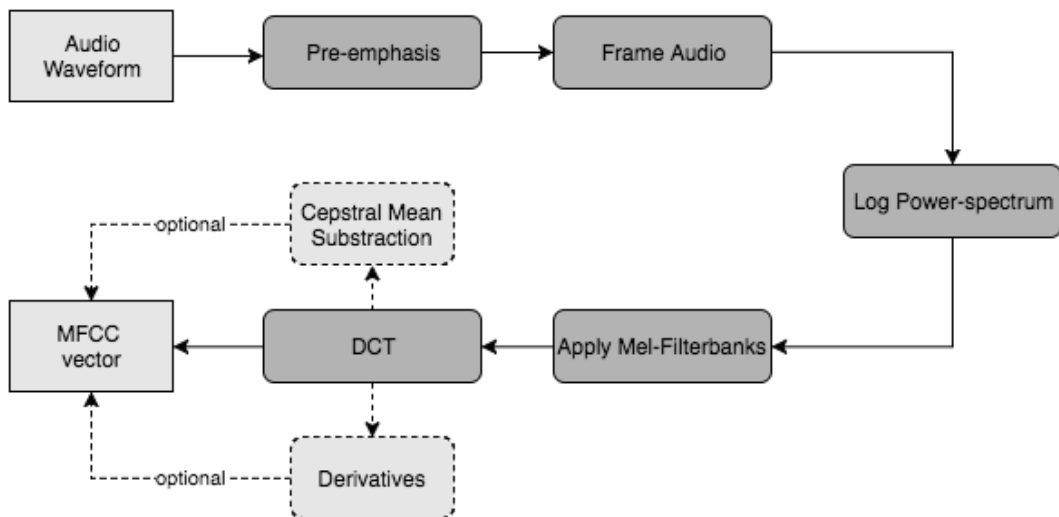


Figure 6.4: MFCC calculation flow diagram.

The MFCC calculation process is shown in **Figure 6.4**. The process can be summarized as follows:

1. Pre-emphasis: the higher frequency information is emphasized.
2. The audio signal is divided up into short frames.
3. The log-power spectrum of each frame is computed.
4. Mel-scaled triangular filterbanks are applied to the power spectrum.
5. The discrete cosine transform (DCT) is applied to the output of the Mel-filterbanks.
6. (Optional) Cepstral Mean Subtraction: Subtract the long-term mean from each coefficient.
7. (Optional) Compute inter-frame derivatives of each coefficient and append these to the already calculated features.

During the pre-emphasis step, the audio signal $x(n)$ is high-pass filtered by a filter with a transfer function of $H(z) = 1 - a * z^{-1}$. This approximately compensates for the natural high frequency suppression of the human vocal tract. When framing the audio, the frame length needs to be short enough so that statistical stationarity can be assumed over the frame, but long enough to provide adequate spectral resolution. The log-power spectrum is computed using **Equations 6.1** and **6.2**.

When applying Mel-filterbanks to the power spectrum, M triangular filters are used in a way similar to the log-filterbanks shown in **Figures 6.2** and **6.3**. However, in this case the filters are not linearly spaced in frequency, but are spaced according to the Mel-frequency

6.2 Feature Extraction

scale. The Mel-scale approximates the frequency resolution of the human auditory system by providing higher resolution at lower frequencies, and lower resolution at higher frequencies. This is motivated by biological and perceptual models of the human cochlea. **Figure 6.5** shows the relationship between linear and Mel-scale frequency, with 10 Mel-scaled triangular filters plotted on the linear frequency axis. To convert from frequency to Mel-frequency, **Equation 6.3** is used.

$$f_{mel}(f) = 2595 \ln\left(1 + \frac{f}{700}\right) \quad (6.3)$$

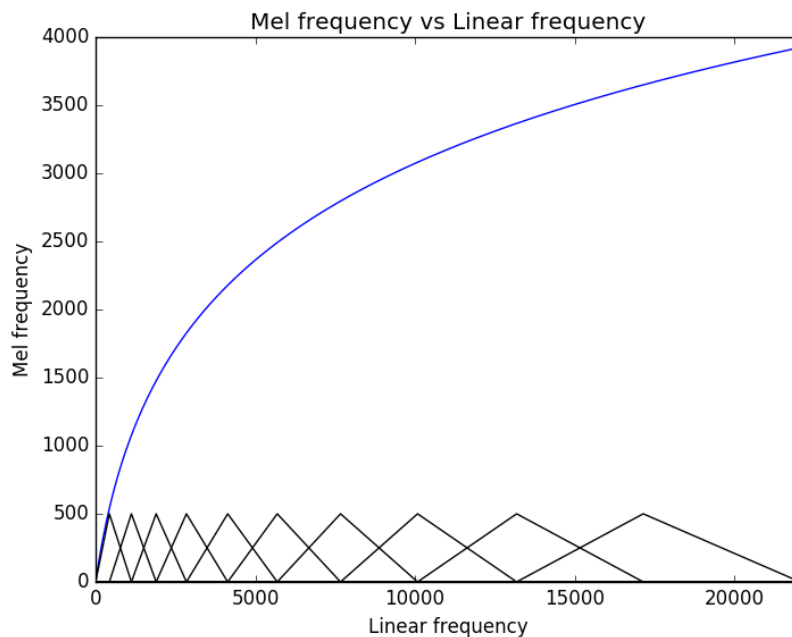


Figure 6.5: Mel-scaled filters

The energies below each Mel-filterbank are summed, resulting in a vector of energies that are processed by the discrete cosine transform (DCT). The DCT acts as a compression step, which decomposes the Mel-filterbanks into real-valued coefficients that can be used to describe the vocal tract envelope. The lower order coefficients represent the overall shape of the envelope while higher order coefficients are related to vocal excitation and noise and are generally discarded. A more in-depth discussion of the DCT can be found in [56].

The DCT can be calculated using **Equation 6.4**, with N the number of data points and k the length of the DCT, usually chosen to equal N .

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n + \frac{1}{2}k\right)\right] \quad k = 0, \dots, N - 1 \quad (6.4)$$

The resulting coefficients are the Mel-frequency cepstral coefficients. As shown in **Figure 6.4**, two additional steps are often included. The MFCCs do not inherently contain any information about the dynamic behaviour of the signal. That is, the coefficients do not tell us how the signal varies over time. As MFCCs are generally used to represent time-varying signals, it would be beneficial if we could include such information in the MFCC vector, especially when working with a model like an HMM that assumes independence between successive frames. This can be achieved by calculating the inter-frame derivatives of each coefficient, using **Equation 6.5**, where d_t represents the derivative at frame t and N the number of frames over which the calculation is performed (typically chosen as $N = 2$).

$$d_t = \frac{\sum_{n=1}^N n(c_{t+n} - c_t - n)}{2 \sum_{n=1}^N n^2} \quad (6.5)$$

A good speech recognition system requires environmental robustness and speaker independence [57]. Being able to perform recognition tasks on audio recorded in different environments, through different recording channels and for any person requires that the recognition system be normalized with respect to these variations. Cepstral mean subtraction (also referred to as cepstral mean normalization) provides a solution to this problem. The basic idea is to subtract the mean from each cepstral coefficient (taken over the entire utterance or even several utterances). When a signal is passed through a linear, time invariant channel, the signal is convolved with the channel transfer function (**Equation 6.6a**). This convolution becomes multiplication when we compute the power spectrum of the signal, as discussed in **Section 6.2.1** and shown here in **Equation 6.6b**. When taking the logarithm of the power spectrum in **Equation 6.6b**, multiplication becomes addition and we obtain the cepstrum in **Equation 6.6c**. Here q refers to the *quefreny*. If we assume that $H[q]$ is constant (our channel does not change while we are recording), we can calculate the mean of $S[q]$, by using **Equation 6.6d**, over all N frames. If we subtract this mean from each frame (**Equation 6.6e**) we obtain $Z_i[q]$, which is free from the effect of $H[q]$.

$$y[n] = x[n] * h[n] \quad (6.6a)$$

$$S[f] = X[f] \cdot H[n] \quad (6.6b)$$

$$S[q] = \log(S[f]) = \log(X[f] \cdot H[n]) = X[q] + H[q] \quad (6.6c)$$

$$\bar{S}[q] = H[q] + \sum_{i=1}^N X_i[q] \quad (6.6d)$$

$$Z_i[q] = S_i[q] - \bar{S}[q] \quad (6.6e)$$

$$= X_i[q] + H[q] - \left(H[q] + \sum_{j=1}^N X_j[q] \right) \quad (6.6f)$$

$$= X_i[q] - \sum_{j=1}^N X_j[q] \quad (6.6g)$$

6.2.3 Zero Crossing Rate (ZCR)

The zero crossing rate of a signal is a measure of how many times the signal moved from a positive to a negative amplitude. ZCR gives an overall indication of how variable the signal is.

ZCR can be defined as:

$$ZCR = \frac{1}{T-1} \sum_{t=1}^{T-1} \Pi(s_t s_{t-1} < 0) \quad (6.7)$$

where the indicator function Π is 1 if the signs of s_t and s_{t-1} differ, else it is 0. Notice that zero values are considered positive in this context.

ZCR provides a feature that can be extraction from a signal in addition to the log filterbank energies or MFCCs and is appended to each feature vector. ZCR was chosen as an additional feature following similar research [38][2][7].

6.2.4 Kurtosis

Kurtosis is used in statistics as measurement of 'peakedness' of a probability density function. Kurtosis can be calculated using the following equation:

$$\mathcal{K}_x = \frac{E[(x_i[k] - \mu)^4]}{\sigma^4} \quad (6.8)$$

In digital signal processing, it is most useful when applied to the power spectrum density function. Kurtosis provides information on the shape of the underlying probability distribution, and can be used as an additional feature to log filterbank energies or MFCCs. The inclusion of kurtosis as a features was motivated by other relevant studies [38][2]. More information about the applications and limitations of spectral kurtosis can be found in [58].

6.3 Experimental Setup

Motivated by the success of MFCCs in many audio classification tasks, we investigate the difference in classifier performance when trained on features that mimic the human auditory system (MFCCs), and on features that do not (log-filterbanks).

Thus our experimental evaluation revolves around training classifiers with either MFCC or log-filterbank features. Both the MFCC and log-filterbank feature vectors were augmented by appending ZCR, kurtosis and log energy values.

6.3.1 Constructing Feature Datasets

Different extraction topologies were investigated as part of the experimental evaluation. Certain hyper-parameters were defined to describe these topologies and are listed in **Table 6.1**. These hyper-parameters are determined by the feature types chosen. The inclusion of the segment parameter S was motivated by the theoretical background of cough sounds.

Table 6.1: Feature extraction hyper parameters.

Variable	Symbol	Description	Range
Frame length	N	Size of frames in samples into which audio is segmented.	256, 512, 1024, 2048, 4096
No. segments	S	Number of segments into which frames were grouped	1,2,3,4
Avg. over segments	A	Do we average the features in each segment?	Binary: Yes/No
No. Filters	F	Number of filters to use when extracting filterbank energies.	40, 60, 80, ..., 200
No. MFCCs	M	Number of lower order MFCC to keep.	13, 26

For all experiments, the DFT length was set to be equal to the frame length. DFTs can be computed efficiently using the FFT when using a length of order $N = 2^k$. Thus the range of frame lengths considered was chosen to be of order 2^k , with $k = 8, 9, 10, 11, 12$.

As shown in **Figure 2.1** and discussed in **Section 2.2.2**, some authors maintain that coughs can be divided into 3 distinct phases, with the second being the most distinctive and carrying the most information regarding the cough characteristics. In order to investigate this concept, we decided to group the frames into S non-overlapping segments to approximately mimic the different phases of a cough. The value of S was empirically chosen, where $S = 1$ is the case where a single segment is used, $S = 2$ is the case when the cough is divided into 2 segments and so forth.

Additionally, we included the option of averaging the feature vectors extracted from each segment. When a frame overlapped the boundary between two segments, it was assigned to the segment to which most of its samples belonged.

The number of filters used when extracting log-filterbank energies (F) was chosen to provide a good frequency resolution (112.5Hz when using 200 filters), without becoming too computationally expensive.

The number of MFCCs (M) was chosen in correspondence with the other studies mentioned in **Section 2.2**. The first and second derivatives (deltas and delta-deltas) were always included and cepstral mean subtraction was always performed during MFCC extraction.

6.4 Experimental Evaluation

In this section we evaluate different classifiers trained on the data acquired through feature extraction on the audio dataset \mathcal{D}_{CC} , described in **Section 4.2**. Three different classifiers were investigated: logistic regression, Gaussian mixture models and decision trees. Logistic regression was used for audio based classification, meta data classification and classifier fusion. GMMs were used for audio based classification, and decision trees were used for meta data classification and classifier fusion.

The classification results are summarized in **Sections 6.4.1, 6.4.2 and 6.4.3**. An overview of the complete evaluation process followed for each classifier is shown in **Figure 6.6**.

Evaluation was performed using nested cross validation. At the top level (A), the dataset is split into 5 folds: 20% of the entire dataset is reserved as an independent test set \mathcal{D}_{test} and 80% is used as a training set \mathcal{D}_{train} . Hyper-parameter optimization was performed using 4-fold cross validation on \mathcal{D}_{train} (B). For each fold, a model was trained for all possible combinations of hyper-parameters, and the best model was chosen, as indicated in **Figure 3.9**. The number of folds to use for cross validation during hyper-parameter optimization was empirically chosen as 4. Higher values did not provide significant increases in model performance. The scoring method used during hyper-parameter optimization was ROC AUC. The hyper-parameters that were optimized for each model are listed in **Table 6.2**. Note that decision trees were only implemented using the clinical database, therefore the maximum tree depth was chosen as the number of clinical features (5).

6.4 Experimental Evaluation

Table 6.2: Values considered for hyper-parameter optimization using grid search.

Classifier	Hyper-parameters	Range
Logistic regression	λ	$10^{-5} - 10^5$
GMM	no. mixes	1-5
	covariance type	diagonal/tied
Decision trees	Splitting criterion	gini/entropy
	Maximum tree depth	1-5

Before training the final model (in each fold), 2-fold cross validation¹ was performed on \mathcal{D}_{train} using the optimal hyper-parameters (C). This step produces probabilities for each sample in the train set. An ROC curve was constructed using these probabilities and an equal error rate threshold γ_{EE} was then chosen as the point on the ROC curve with the smallest difference between the TPR and the FPR. This threshold value γ_{EE} was later used during model evaluation on \mathcal{D}_{test} . The classifier was then trained on the entire \mathcal{D}_{train} with the optimal hyper-parameters.

Two evaluation methods were investigated. Conventionally, the testing data is presented to the trained model and the probability of each sample belonging to either class is calculated. Classification can be carried out on a frame-by-frame basis by subjecting the class-conditional probability associated with each frame to a threshold γ . Intuitively, this provides a result for each frame of every cough in the dataset. However, we are ultimately not interested in the frame accuracies. Rather, we want to distinguish between patients who have TB and those who do not. A more useful result would thus be one classification (or diagnosis) per patient, which takes into account all frames in all coughs available for that patient.

Therefore, *patient-based* classification results were computed by calculating a specially developed metric referred to as the *Tuberculosis Index Score* (TIS). In order to calculate the TIS, we calculate the mean probability of all frames being TB positive: $P(Y = 1|X, \theta)$ for all frames $X = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_j\}$ in a cough with j the number of frames, given the trained model parameters θ , and $Y = \{0, 1\}$ representing the class to which the data in X belongs, with 1 indicating TB positive and 0 indicating TB negative. The probability of a frame being TB positive is defined in **Equation 3.13** for a logistic regression classifier and in **Equation 3.30** for GMMs. When converting the likelihoods into posterior probabilities in **Equation 3.30**, the prior probabilities were taken to be equal, because we are using ROC AUC as the main evaluation metric and ROC inherently performs the task of varying these prior probabilities by adjusting the threshold value γ .

¹The number of folds was chosen as 2 for fast computation time. Choosing higher values did not provide better results.

6.4 Experimental Evaluation

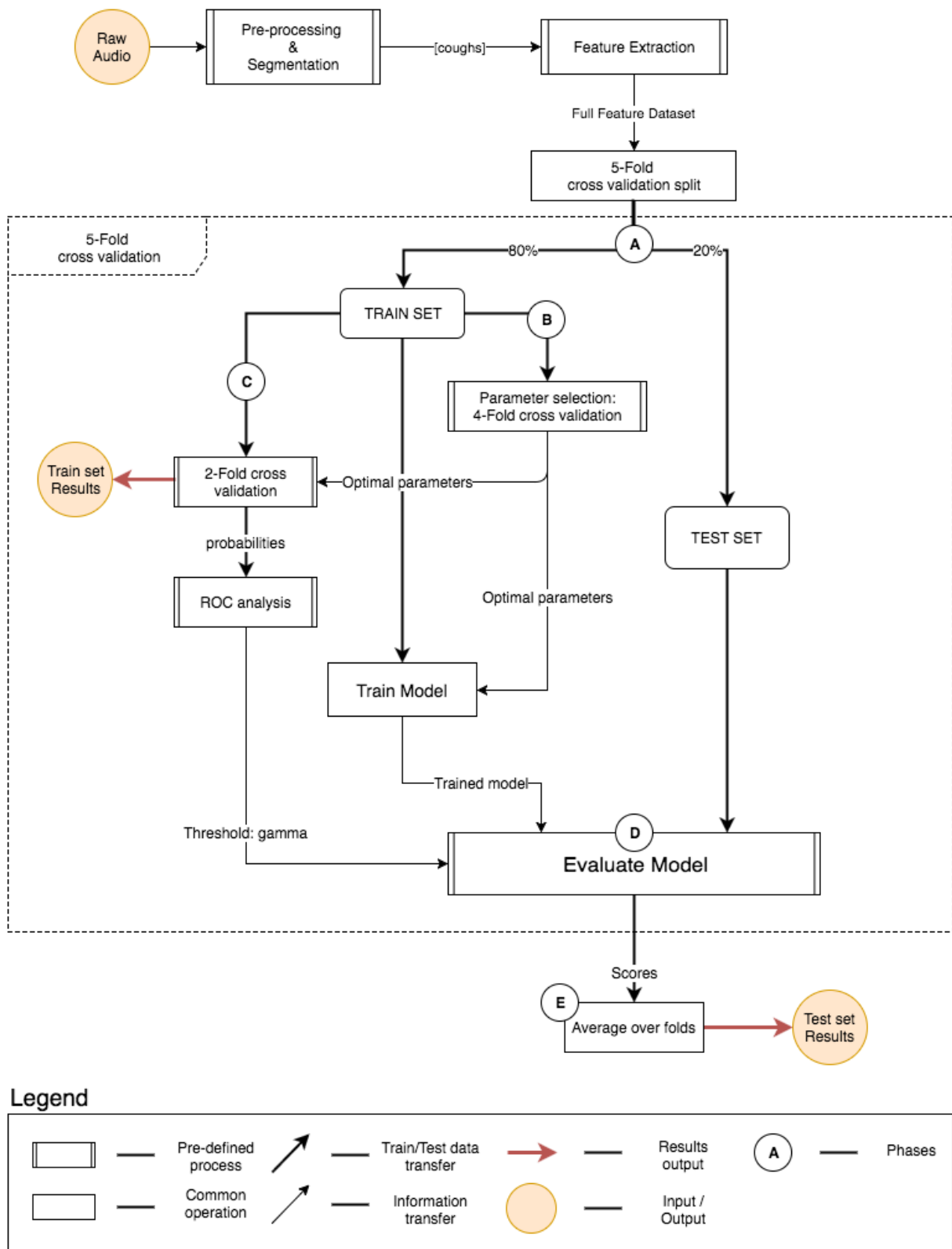


Figure 6.6: An overview of the classification and evaluation process.

6.4 Experimental Evaluation

If the mean probability of a cough being TB positive (\bar{P}) is above the equal error rate threshold γ_{EE} , the cough is labelled as TB positive. The value of \bar{P} is calculated by summing all the posterior probabilities of each frame in that cough and dividing by the number of frames, shown in **Equation 6.9** with K the number of frames in a specific cough. The TIS is then calculated using **Equation 6.11**, with $\sum_{i=1}^N Q$ the number of TB positive coughs by a patient, defined by **Equation 6.10** and N the total number of coughs by that patient.

$$\bar{P} = \frac{\sum_{i=1}^K P(Y = 1|X, \theta)}{K} \quad (6.9)$$

$$Q = \begin{cases} 1 & \text{if } \bar{P} \geq \gamma_{EE} \\ 0 & \text{otherwise} \end{cases} \quad (6.10)$$

$$TIS = \frac{\sum_{i=1}^N Q}{N} \quad (6.11)$$

Thus, the TIS is the proportion of all coughs by a certain patient that are classified as TB positive. A patient is diagnosed as being TB positive, if more than half of their coughs were classified as TB positive, or when $TIS > 0.5$. The TIS is inspired by the PCI metric proposed in [38], but extended to consider γ_{EE} when classifying each cough.

One disadvantage of the TIS as an evaluation method is that if a patient provides few coughs, the value can become misleading. For instance if a patient has 5 coughs, the TIS would require at least 3 coughs to be classified as TB positive in order to make a positive diagnosis. Thus, in this example, it requires 60% of the coughs to be TB positive and therefore provides a coarse resolution.

Therefore, in addition to the TIS, a second evaluation method referred to as the *Average Diagnosis Score* (ADS) was introduced. The ADS calculates the average probability $P(Y = 1|X)$ over all frames of all coughs by a patient and performs a diagnosis by comparing this probability to γ_{EE} . The ADS is calculated using **Equation 6.12**, where N is the total number of frames in all coughs by a specific patient.

$$ADS = \frac{\sum_{i=1}^N P(Y = 1|X)}{N} \quad (6.12)$$

As shown in **Figure 6.6**, for each fold the model was evaluated on the test set and the scores were averaged over all folds to produce the final scores. During model evaluation, the sensitivity, specificity, accuracy and Cohen's kappa of the model were computed for both ADS and TIS. In **Section 3.4.3** we defined the area under the ROC curve (AUC) as an important evaluation metric. To compute the ROC AUC for both TIS and ADS, the true positive rate (TPR) and false positive rate (FPR) were calculated in each fold and

6.4 Experimental Evaluation

averaged over all folds to obtain the mean TPR and FPR. The mean TPR and FPR were then used to construct an ROC curve and the AUC was calculated using that curve. For TIS, the TPR and FPR were computed using the ratio in **Equation 6.11** as an evaluation value, and for ADS the average probability over all frames (**Equation 6.12**) was used.

After 5-fold cross validation, the following scores were produced for each evaluated classification system: sensitivity, specificity, accuracy, Cohen's kappa and AUC. Note that the accuracy is calculated at the equal error rate, using the decision threshold γ_{EE} as selected during validation on the training set. In this way, the thresholds to compute the ADS and TIS accuracies in each fold were not optimized on the test set.

To summarize, two methods were used to evaluate the model performance: the *Tuberculosis Index Score* (TIS) and the *Average Diagnosis Score* (ADS). Through these two methods, five metrics were calculated: sensitivity, specificity, accuracy, AUC and Cohen's kappa. The AUC values were computed using ROC curves, which were constructed by comparing the TIS and ADS values to a decision threshold γ which is varied over the range 0 – 1. For the ADS method, the sensitivity, specificity, accuracy and kappa were computed using the threshold value which minimized the difference between sensitivity and specificity, referred to as the equal error rate threshold (γ_{EE}). The equal error rate threshold was determined using 2-fold cross validation on the training set. For the TIS method, these metrics were computed using the threshold value $\gamma = 0.5$.

6.4.1 Classification Performance

In this section we report on the performance of classifiers trained as described in the previous section. The same training, optimization and evaluation procedures were followed in each case. Because of the high number of possible feature extraction parameter combinations, the best results for each number of filters (F), and number of MFCCs (M) are reported.

The best results were selected using the AUC values.

The results achieved using a logistic regression classifier are summarized in **Table 6.3**. Note that, where $A = \text{False}$, the value of S is not applicable, because S is only taken into consideration when averaging features over each segment.

6.4 Experimental Evaluation

Table 6.3: Logistic regression patient-level results. Refer to Table 6.1 for clarification of the abbreviations used.

Config	Feature Parameters				ADS Results					TIS Results				
	Feature	N	S	A	Sens	Spec	Acc	Kappa	AUC	Sens	Spec	Acc	Kappa	AUC
1	F:40	256	4	TRUE	0.517	0.850	0.693	0.565	0.770	0.517	0.900	0.722	0.615	0.768
2	F:60	256	3	TRUE	0.517	0.900	0.722	0.589	0.784	0.567	0.900	0.747	0.618	0.807
3	F:80	256	3	TRUE	0.567	0.900	0.747	0.618	0.791	0.500	0.900	0.718	0.590	0.769
4	F:100	1024	-	FALSE	0.433	0.950	0.718	0.588	0.777	0.433	0.900	0.690	0.538	0.748
5	F:120	2048	2	TRUE	0.417	0.950	0.718	0.571	0.809	0.417	0.900	0.690	0.541	0.640
6	F:140	2048	-	FALSE	0.417	0.950	0.715	0.572	0.812	0.417	0.950	0.715	0.572	0.779
7	F:160	2048	2	TRUE	0.417	0.950	0.715	0.576	0.805	0.483	0.900	0.715	0.586	0.792
8	F:180	1024	2	TRUE	0.067	0.900	0.525	0.336	0.795	0.433	0.950	0.718	0.564	0.771
9	F:200	2048	1	TRUE	0.617	0.950	0.800	0.703	0.805	0.667	0.810	0.743	0.632	0.764
10	M:13	1024	-	FALSE	0.400	0.820	0.633	0.455	0.711	0.400	0.820	0.633	0.455	0.656
11	M:26	2048	2	TRUE	0.483	0.720	0.623	0.468	0.634	0.617	0.720	0.676	0.553	0.628

From this table, we can see that the best ADS AUC of 81.2% is achieved with $F = 100$, $N = 2048$ and $A = \text{False}$. A corresponding accuracy of 71.5% with sensitivity and specificity of 41.7% and 95.0% is achieved with these parameters. An AUC value higher than the accuracy indicates that the model would be able to achieve a higher accuracy if the decision threshold γ was optimized. However, as mentioned earlier, the accuracy is computed at the equal error rate determined during validation on the training set. A κ of 0.572 indicates that the accuracy of the model is 57.2% of the way between the expected accuracy and 100%.

The best ADS *accuracy* of 80% is achieved with $F = 200$, $N = 2048$, $S = 1$ and $A = \text{True}$. The AUC with these parameters is 80.5%, which indicates that predictions were made at a threshold value that adequately represents the models overall performance. The corresponding sensitivity, specificity and κ are 61.7%, 95.0% and 0.703. **Figure 6.7** shows the mean ROC curve computed over 5-folds of the best performing model in terms of ADS AUC, with the TIS based ROC curve for that model shown on the same graph. The blue dotted line represents a classifier making random guesses.

The TIS evaluation measure only produces better results than ADS in one instance ($F = 60$, $N = 256$, $B = 3$, $A = \text{True}$), with an AUC of 80.7% and corresponding sensitivity, specificity, accuracy and κ of 56.7%, 90.0%, 74.5% and 0.618. Compared to an ADS AUC of 78.4%, this represents only a marginal improvement and thus indicates that the ADS is a better criterion to use in model optimization. **Figure 6.8** shows the mean ROC curve of the best performing model in terms of TIS AUC, with the corresponding ADS based ROC curve shown on the same graph.

We can see from these results that, overall, the model achieves a high specificity while achieving a moderate and sometimes poor (6.7% at $F = 180$) sensitivity when evaluated using the value of γ_{EE} determined during the validation process.

6.4 Experimental Evaluation

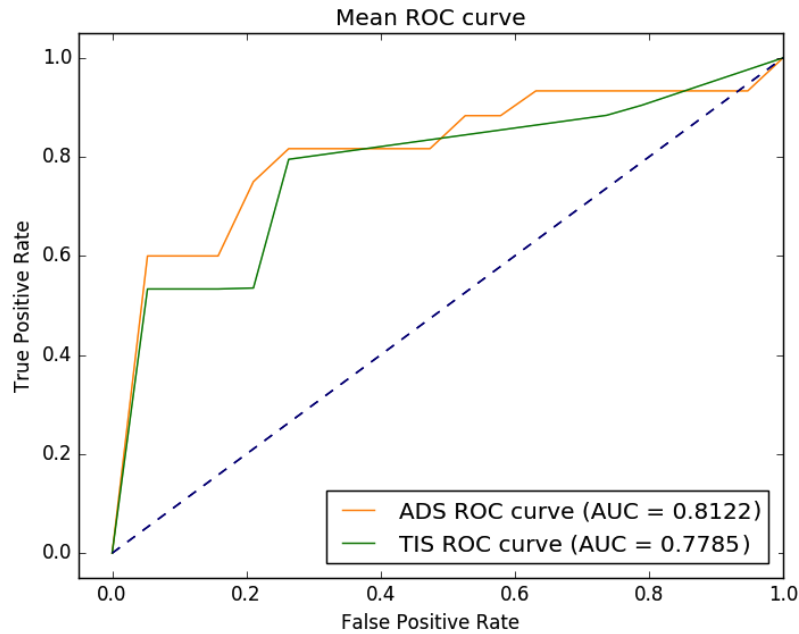


Figure 6.7: Mean ROC curve of the best performing logistic regression model in terms of ADS AUC with the TIS based curve included.

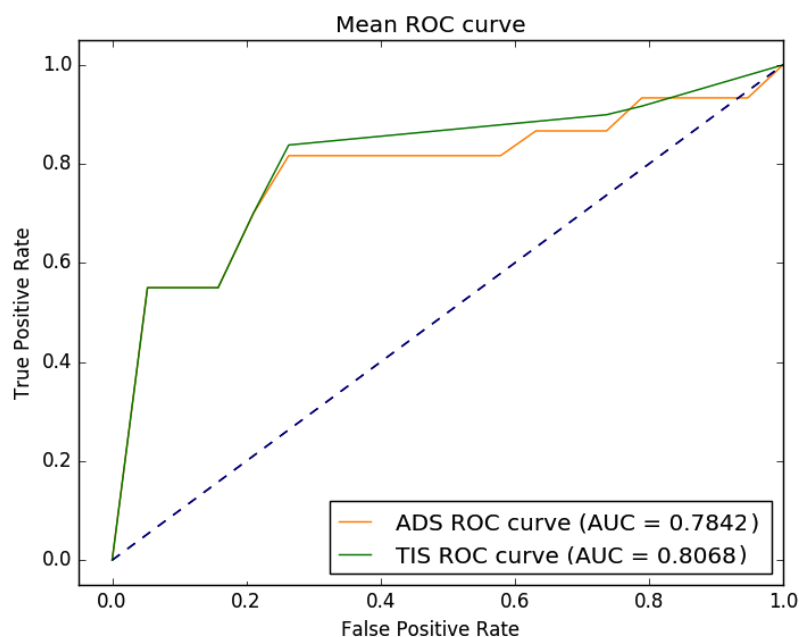


Figure 6.8: Mean ROC curve of the best performing logistic regression model in terms of TIS with the ADS based results included.

It is important and interesting to note the difference in performance between MFCC- and filterbank-based models. The best performing MFCC-based model has a reported ADS AUC of 71.1% and corresponding sensitivity, specificity, accuracy and κ of 40.0%, 82.0%, 63.3% and 0.468 respectively. The worst performing filterbank-based model as an

6.4 Experimental Evaluation

ADS AUC of 77% and corresponding sensitivity, specificity, accuracy and κ of 51.7%, 85.0%, 69.3% and 0.565 respectively.

From this we can conclude that using filterbank features is more effective when classifying cough sounds. This suggests that the reduction in spectral resolution based on models of the human auditory system that is used by the MFCCs discards information that is useful for classifying cough sounds. One can speculate that the system is basing its decisions on audio phenomena in the recorded sound that are not perceivable by a human listener.

Table 6.4 shows the results achieved using a GMM classifier. Overall the results are worse than those achieved by the logistic regression classifier.

The best ADS AUC of 79.2% is achieved with the feature extraction parameters of $F = 60, N = 4096, S = 4$ and $A = \text{True}$, with a corresponding accuracy of 53.2% and a sensitivity and specificity of 60.0% and 40.0% respectively. The difference between accuracy and AUC of this model indicates that predictions were made at a poor threshold value γ_{EE} . The best TIS based results are achieved at the same feature extraction parameters, with an AUC of 76.1% and a corresponding accuracy, sensitivity, specificity and κ of 63%, 46.7%, 73.0% and 0.445 respectively.

The ROC curves of the best performing GMM models are shown in **Figure 6.9**.

Table 6.4: Gaussian mixture model patient-level results. Refer to Table 6.1 for clarification of the abbreviations used.

Config	Feature Parameters				ADS Results					TIS Results				
	Feature	N	S	A	Sens	Spec	Acc	AUC	Kappa	Sens	Spec	Acc	AUC	Kappa
1	F:40	512	4	TRUE	0.667	0.400	0.532	0.607	0.367	0.600	0.450	0.532	0.558	0.357
2	F:60	4096	4	TRUE	0.400	0.600	0.532	0.792	0.346	0.467	0.730	0.630	0.761	0.445
3	F:100	4096	4	TRUE	0.400	0.600	0.532	0.737	0.346	0.467	0.680	0.605	0.664	0.421
4	F:120	4096	4	TRUE	0.600	0.400	0.503	0.622	0.334	0.667	0.530	0.601	0.514	0.432
5	F:140	1024	-	FALSE	0.550	0.450	0.503	0.544	0.334	0.617	0.500	0.557	0.520	0.392
6	F:160	1024	-	FALSE	0.500	0.450	0.478	0.544	0.313	0.550	0.500	0.528	0.468	0.358
7	F:180	4096	-	FALSE	0.400	0.600	0.503	0.562	0.334	0.400	0.600	0.503	0.519	0.334
8	F:200	256	1	TRUE	0.600	0.400	0.475	0.535	0.321	0.467	0.450	0.446	0.405	0.294
9	F:80	4096	4	TRUE	0.350	0.650	0.532	0.737	0.346	0.467	0.780	0.655	0.671	0.474
10	M:13	4096	3	TRUE	0.867	0.350	0.589	0.528	0.445	0.733	0.400	0.557	0.554	0.395
11	M:26	4096	3	TRUE	1.000	0.000	0.446	0.566	0.308	1.000	0.250	0.585	0.667	0.437

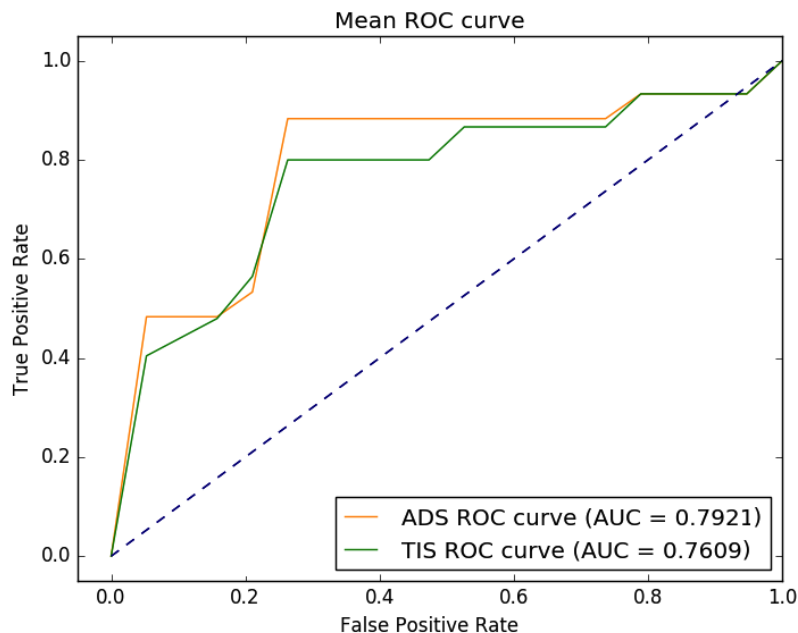


Figure 6.9: Mean ROC curve of best performing GMM model in terms of ADS and TIS evaluation. The best ADS and TIS results are obtained from the same feature extraction parameters.

6.4.2 Meta Data Classifier

Two classifiers were trained on the clinical dataset described in **Section 4.1.3**. The classifier training procedure and model evaluation was similar to that followed for the audio classifiers. The evaluation methods ADS and TIS are not applicable in this case and were not used. The data consists of one sample per patient and conventional (sample-based) methods were used.

Table 6.5 shows the classification results achieved for this dataset. The results are split into two groups: reduced dataset and full dataset. The reduced dataset refers to the results achieved when using the subset of the clinical dataset consisting only of the patients for which we have audio recordings. The full dataset represents the results achieved on the full clinical dataset.

The reduced dataset results can be directly compared to the results on the audio dataset, while the full dataset results are included as supplementary information to indicate how the classifier performs when given more data.

From **Table 6.5** we can see that the logistic regression classifier outperforms the decision tree classifier. Furthermore, the logistic regression model trained on the audio data outperforms the logistic regression model trained on the clinical data with a reported AUC of 81.20% compared to the AUC of 77.36% obtained on the clinical dataset. The

6.4 Experimental Evaluation

reported AUC for a logistic regression classifier using the full dataset is 79.50%, which is still lower than the best performing audio based classifier.

This indicates that, with respect to TB diagnosis, the audio data appears to be more information rich than the objective clinical measurements at our disposal. These results are a good indication that using audio data as a diagnosis tool for TB is a viable method in resource poor areas.

Table 6.5: Meta-classifiers results

Classifier Type	Reduced Dataset					Full Dataset				
	Sens	Spec	Acc	AUC	Kappa	Sens	Spec	Acc	AUC	Kappa
Logistic regression	0.6833	0.81	0.7556	0.7736	0.62217	0.716	0.6943	0.71	0.795	0.4709
Decision trees	0.25	0.826	0.5644	0.5271	0.3976	0.515	0.7464	0.5697	0.7212	0.3627

6.4.3 Classifier Fusion

Now that we have used both the audio and clinical data to perform classification, we can combine the output of the two best models by classifier fusion. The two best models that will be combined are the best models reported in **Tables 6.3** and **6.5**.

There are various methods for fusing classifiers [59], however in this study we will focus on 3 non-learning methods and 2 learning methods. A non-learning method uses general predefined rules to combine the output of multiple classifiers, while learning methods use the output of multiple classifiers as features for another learning algorithm.

For non-learning methods, we used weighted voting, the sum rule and a modified version of the product rule. For learning methods we used logistic regression and decision trees.

The following features were used for classifier fusion:

- The probability of a sample being TB positive. For the audio classifier we used the ADS value of each patient, and for the meta classifier we used $P(Y = 1|X)$ of each sample.
- The predictions made by the meta- and audio classifiers (0 for TB negative and 1 for TB positive).
- The equal error rate decision threshold values γ_{EE} for the meta- and audio classifiers at which the predictions were made.

Weighted voting makes a prediction by picking the class with the highest score, after applying a certain weight to the prediction of each classifier and then summing all the predictions per class. The voting weights used in this study are calculated with **Equation 6.13**, where w_k is the weight of classifier k , a_k is the accuracy of that classifier on the training set and C is the number of classifiers by the model accuracy on the training set

6.4 Experimental Evaluation

[59]. Hence, weighted voting gives a higher weight to models that are more accurate on the training corpus.

$$w_k = \frac{a_k}{\sum_{j=1}^C a_j} \quad (6.13)$$

After calculating the weights, the prediction $\hat{c}(\mathbf{x}_i)$ for each sample \mathbf{x}_i can be calculated using **Equation 6.14**. Here $\hat{c}_k(\mathbf{x}_i)$ is the prediction of classifier C_i and $\delta(A, B) = 1$ if $A = B$, otherwise 0.

Since $\sum_{k=1}^K w_k = 1$ the sum in **Equation 6.14** can be interpreted as the probability of sample \mathbf{x}_i being classified as c_j . This probability was used to compute the ROC AUC value.

$$\hat{c}(\mathbf{x}) = \underset{c_j \in C}{\operatorname{argmax}} \sum_k w_k \delta(\hat{c}_k(\mathbf{x}_i), c_j) \quad (6.14)$$

The sum rule computes the sum of the per-class posterior probabilities for each classifier and a prediction is made by choosing the class with largest sum of probabilities. This is equivalent to weighted voting with equal weights.

Similarly, the product rule computes the sum of the per-class probabilities. However, instead of making a prediction by choosing the class with the largest sum, we used the product of the equal error rates of each classifier ($\gamma_{EE1}, \gamma_{EE2}$) as the new threshold value. Furthermore, we used the product of probabilities for computing the ROC AUC.

Table 6.6 shows a summary of the results of all the tested classifier fusion techniques. We can see that the best AUC is achieved when using the sum rule, with a reported AUC of 84.31%. Notably, most methods achieved a higher AUC than the audio-based model (81.12%) and the meta-classifier (77.36%) separately.

This indicates that combining the audio based diagnosis with objectively measured clinical data can improve the effectiveness of the diagnosis system.

Table 6.6: Results of classifier fusion.

Method	Sens	Spec	Acc	AUC	Kappa
Non-Learning Methods					
Weighted voting	0.4118	0.9524	0.7105	0.8053	0.5381
Sum rule	0.8235	0.3809	0.57891	0.8431	0.4142
Product rule	0.6471	0.9048	0.7895	0.8319	0.6457
Learning Methods					
Logistic regression	0.7058	0.8095	0.7632	0.8406	0.6127
Decision trees	0.3529	0.7142	0.5526	0.5230	0.3710

6.5 Discussion and Further Investigation

In this section, we discuss and interpret the results reported in this chapter. Additionally, we investigate which parts of the frequency spectrum provides the most information by performing feature selection.

As will be shown later in this section, due to our small dataset our results contain a significant amount of variance. This means that the optimal models mentioned in the previous sections might not produce optimal results given a new dataset. For that reason, we will not discuss why certain feature extraction parameters produced the best results - because it might be that different extraction parameters would produce better results on a new dataset. However, the best performing models were chosen for the purpose of further investigation, knowing that some variance is expected.

In **Table 6.3** we showed that the filterbank energy features achieved higher accuracies than MFCC features. This might be due to the low frequency resolution of MFCCs in the upper part of the spectrum. Following this observation, we investigate which parts of the frequency spectrum provide the most information for classification. The top performing configurations in **Table 6.3** (configuration 6) were used for this investigation. The training and evaluation was done according to the same procedure used to generate **Table 6.3**.

The frequency spectrum was divided into 20 non-overlapping segments and a logistic regression classifier was then trained and evaluated independently for each of the 20 frequency bands. The frequency spectrum spans 0 – 22.1kHz, thus each segment contains filterbank energies for a frequency range of approximately 1105Hz. For example, if $F = 100$, then each 1105Hz segment contains $\frac{F}{20} = 5$ filterbanks. The first 5 filterbanks are selected and used for training and testing, then the next 5 and so on. This is repeated for all 20 segments, and the AUC results are plotted for each segment. For different values of F , $\frac{F}{20}$ does not always produce a frequency resolution of precisely 1105Hz, but for display purposes this was not indicated on the figures.

The results (in terms of individual frequency bands) for the best performing model in **Table 6.3** is shown in **Figure 6.10**. We see that there is a significant performance gain when the system is trained on the features within the frequency band 0-1105Hz. This is interesting because human speech sounds fall mostly into the frequency range of 300-3000Hz, and the range of vowel sounds are mostly below 1000Hz, as shown in **Figure 6.11**.

6.5 Discussion and Further Investigation

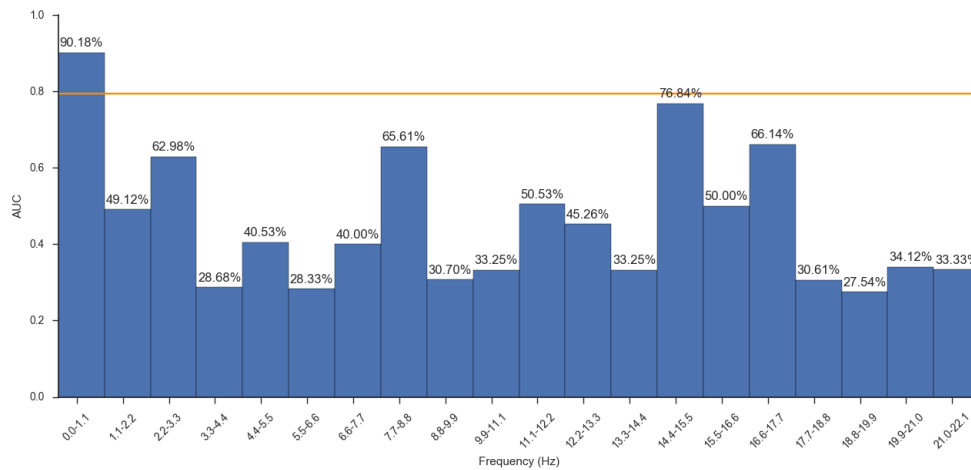


Figure 6.10: Individual frequency band ADS AUC results for F:140.

As mentioned in **Section 2.2.2**, coughs can be divided into three phases, with the second being identified by some studies as the most information rich. When observing our data, we found that the cough patterns differ significantly between patients. However, the second phase is considered as the exhalation phase, and is thus less affected by a patient's unique way of expelling foreign substances or chemicals during coughing.

It is possible that the data in the frequency band 0-1105Hz yields a better performance, because the data originates mostly from the exhalation phase of coughs and thus carries the most vital information for classifying the nature of coughs.

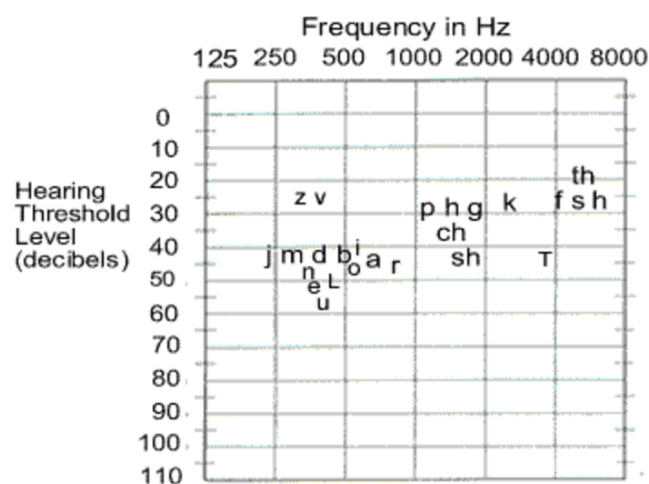


Figure 6.11: Vowel and consonant fundamental frequency ranges (reproduced from [60]).

Figure 6.12 shows the AUC calculated for values of F between 40 and 200, using

6.5 Discussion and Further Investigation

selective frequency band features. From these graphs we can observe that the best results are achieved by all models when isolating the data from the frequency band 0 - 1105Hz. Note that these models were all evaluated without the augmented features (ZCR, kurtosis, log-energy).

Another observation from the results in **Figure 6.12** is the peak in AUC around the frequency bands of 8kHz (7.7kHz - 8.8kHz) and 15kHz (14.3kHz - 15.5kHz). These same peaks are visible for most classifiers in **Figure 6.12**. We can speculate that within those frequency bands lies spectral information indicative of TB, possibly audio artefacts of the physical deterioration of the lung tissue (cavities, nodules, etc), causing high frequency spectral content. Alternatively, this could be caused by fricative (consonant) sounds that TB patients are more prone to. It is difficult to confirm the reason for these performance peaks, and is left for future research.

Before further discussing the observations mentioned above, we will consider the variance of the results in order to determine whether differences in performance in individual frequency bands are statistically significant. We used bootstrapping as described in [61] to estimate the standard deviation in each frequency band. The overall idea of the bootstrapping method is to take samples (with replacement) from the pool of results to simulate multiple, smaller sets of results. The means over all bootstrap samples are then taken as the reported results and the mean squared error over all bootstrap samples is used as the variance. This allows both the mean and standard deviation of the ADS AUC to be estimated. The number of bootstrap samples (B) was set to 100. Using higher values for B ([61] suggests choosing B around 10^3) did not provide substantially different results.

Figure 6.13 shows the results of the same data represented in **Figure 6.10** but now evaluated using the bootstrap. The red vertical bars represent the standard deviation the AUC of each frequency band. The dotted horizontal orange lines show the standard deviation of the model trained on all filterbank energy data. From this we see that the model has a standard deviation of approximately 10% when trained on all the filterbank features.

While the statistical significance cannot be claimed on the differences in ADS AUC values, the overall trends in **Figure 6.12** and **6.13** indicate that some frequency bands are consistently more useful for classification than others. A larger dataset is needed to reduce the statistical uncertainties about the averages in order to determine which differences are significant, however.

6.5 Discussion and Further Investigation

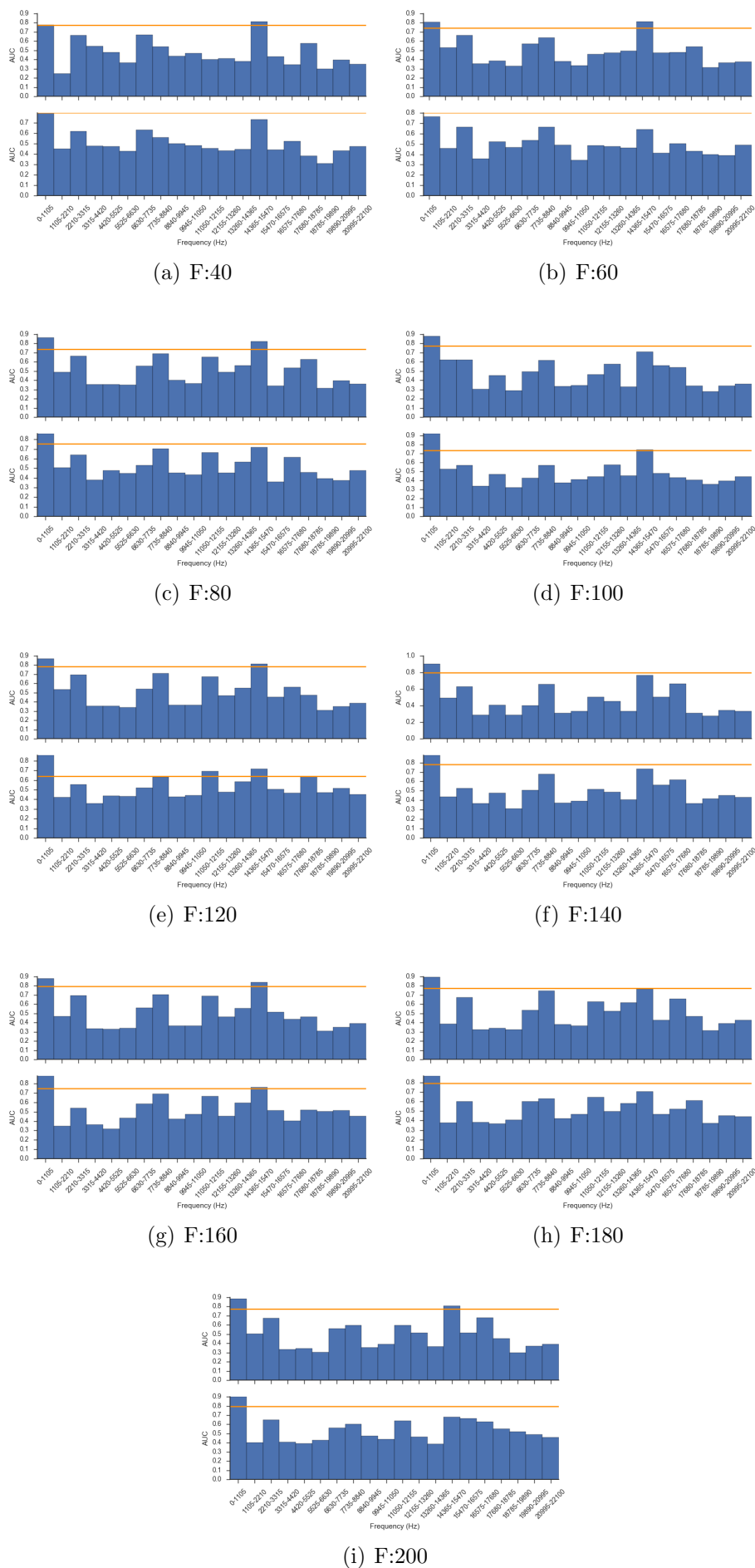


Figure 6.12: Classification performance on filterbank segments.

6.5 Discussion and Further Investigation

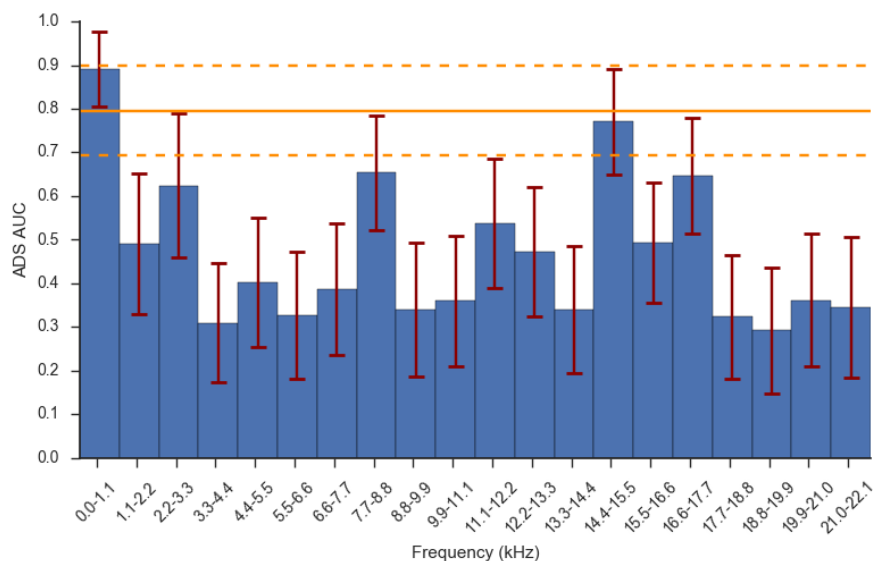


Figure 6.13: Individual frequency band ADS AUC results for F:140 with bootstrap means and standard deviations.

Next, we investigate the effects of feature selection on the best model (configuration 6 in **Table 6.3**). We used two feature selection approaches: greedy search and forward selection [62]. The greedy search algorithm first estimates the model performance on each feature individually, and then sorts the features based on performance (AUC). The features are then recursively included during training, until all features are used. A disadvantage of the greedy search algorithm is that it is prone to find sub optimal results, because the sequence in which features are added is only determined by their individual performance and not by how they would compliment each other. The forward search algorithm, albeit computationally more expensive, addresses this to a large extent. The feature with the best individual performance is combined with all other features, and the best combination is retained. Then all other features are combined with that combination, until all features are used. Note that the true optimum feature combination can only be found with an extensive grid search.

The results for the greedy search algorithm are shown in **Figure 6.14**. The standard deviation is shown for every 5 features added to the training set for display purposes. An initial spike in performance is observed, followed by a plateau and then a fast climb in performance as the last features are added. The initial spike in performance could be attributed to the fact that the features are sorted by individual performance and added in that sequence. Similarly, the plateau in performance is most likely due to the addition of features that performed neither well nor poorly. The final spike in performance could be due to features that perform poorly individually (and are thus added last) complementing the classification capabilities of the best performing features.

6.5 Discussion and Further Investigation

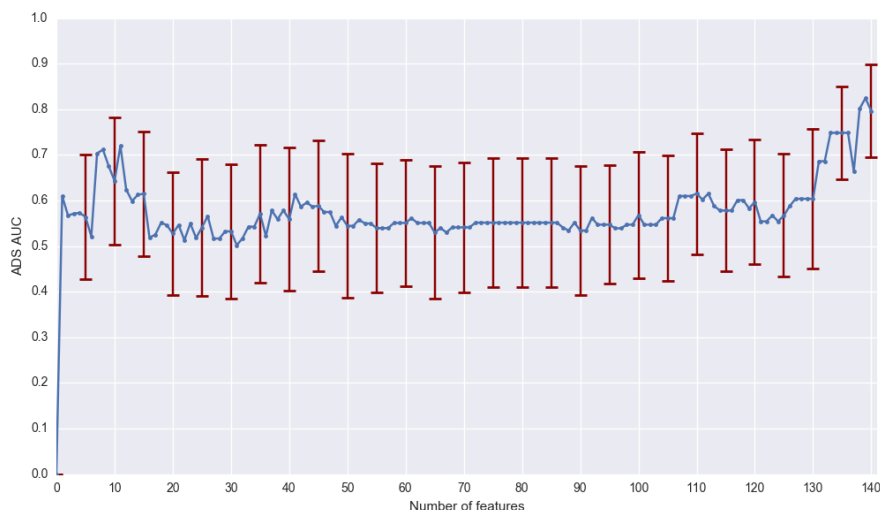


Figure 6.14: Performance in terms of ADS AUC for greedy search feature selection.

The forward search algorithm results are shown in **Figure 6.15**. We can observe that the model performance reaches a plateau after the first 5 features, with a reported AUC of 94.94% and a standard deviation of 4.62%. The 5 filterbanks that produce this result, in order of performance gain, are centered around the frequencies: 236, 550, 10418, 79 and 4894 Hz. This shows that utilizing data from the upper part of the frequency spectrum (10418 Hz), with a focus on the frequencies below 1000Hz provides the best results. This supports our previous findings that MFCC-based models, with a low resolution at higher frequencies, performed worse than log-filterbank based models. We also note that the standard deviation becomes larger when more features added to the model. A smaller standard deviation means that the model has generalized better on the test set since its performance is more consistent.

In contrast to the greedy search algorithm, we see that the performance decreases and standard deviation increases as more features are added. We observe that for the greedy search algorithm the standard deviation remained approximately constant for all combinations of features, indicating that the model did not generalize as well on the test set.

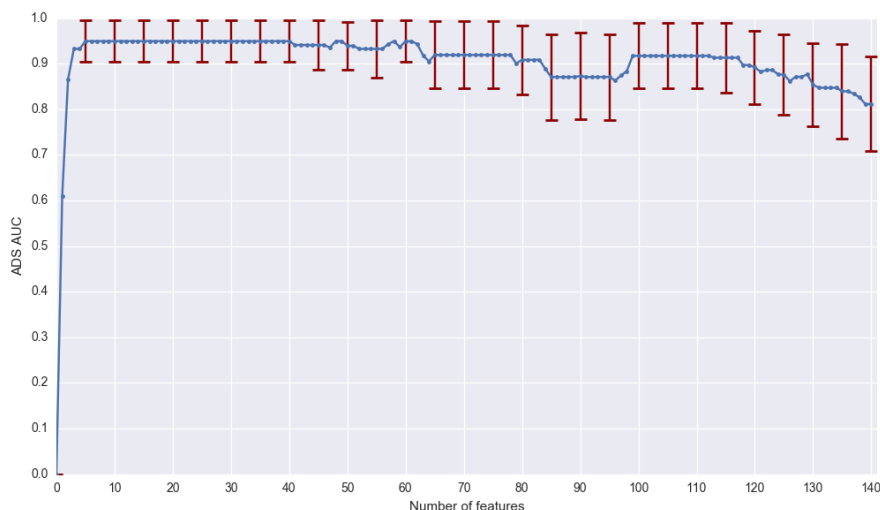


Figure 6.15: Forward selection algorithm results.

For the TIS criterion, similar forward search and grid search results were obtained, but on average the performance was approximately 10% lower.

6.6 Summary

In this chapter we have presented the evaluation of different classifiers and discussed the achieved results. Different feature extraction methods were described and feature extraction topologies were explored. The difference in performance between filterbank energy features and MFCC features was investigated. It was shown that filterbank energy based features outperform MFCC based features, which seems to indicate that the cough audio contains information relevant to TB diagnosis that is not perceivable by the human auditory system.

The performance of a logistic regression classifier and a Gaussian mixture model (GMM) on audio data was compared. The logistic regression classifier outperformed the GMM, with a reported AUC of 81.2% versus 79.2%.

The classification accuracies obtained using the audio data was then compared to those obtained using objective clinical measurements. Logistic regression and decision tree (DT) classifiers were evaluated on the clinical data. The logistic regression classifier outperformed the DT with a reported AUC of 77.36% versus 52.71% using only the clinical data from patients who were present in the audio dataset. The best clinical and audio classifiers were subsequently combined using various classifier fusion techniques. The best fusion results were obtained by using the sum rule of the posterior class probabilities, with a reported AUC of 84.31%.

We then investigated which frequency bands provide the most information for classification. The frequency spectrum was divided into 20 equal segments and a logistic regression classifier was evaluated on features from each segment individually. We found that using features from a limited frequency band provided a considerable increase in AUC. Furthermore, by means of the forward search feature selection algorithm we were able to identify optimal feature combinations. The best performing system designed in this way exhibited an AUC of 94.94% with a standard deviation of 4.62%. The sensitivity, specificity, accuracy and kappa of this model are 61.67%, 91.00%, 78.77% and 0.6511 respectively.

In **Figure 6.13**, the AUC of the same model using all filterbank features is indicated as approximately 79.89% with a standard deviation of 10.0%¹.

The mean of the model trained on all features (79.89%) is approximately 3 standard deviations away from the mean obtained using the best 5 features ($3 \times 4.62\% = 13.86\%$ and $79.89\% + 13.86\% = 93.75\% < 94.94\%$). Assuming a Gaussian distribution for the bootstrap samples, the model trained on the 5 best features improves on the mean accuracy of the model trained on all features with approximately 99% confidence. In addition and importantly, the standard deviation of the AUC of the model trained on the 5 best features is lower than that of the model trained on all features, indicating that this model is more consistent in its classification.

Therefore, using the top 5 performing features after forward search feature selection we obtain a significantly and consistently better system than when training on all features.

Table 6.7 shows the performance achieved by a combination of this best audio classifier (after feature selection) with the best clinical data based classifier. We see that a logistic regression classifier is able to best fuse the outputs of the audio and meta classifiers, with a sensitivity, specificity, accuracy, AUC and kappa of 82.35%, 80.95%, 81.58%, 94.34% and 0.6867 respectively. Although the AUC decreases with 0.5%, all other metrics improved, which indicates the model has generalized better on the test set.

Table 6.7: Classifier combination methods results for model trained on 5 best features.

Method	Sens	Spec	Acc	AUC	Kappa
Weighted voting	0.6471	0.8095	0.7368	0.7885	0.5778
Sum rule	0.8235	0.3810	0.5789	0.8431	0.4143
Product rule	0.6471	0.9048	0.7895	0.8319	0.6457
Logistic regression	0.8235	0.8095	0.8158	0.9434	0.6867
Decision trees	0.8235	0.7143	0.7632	0.8210	0.6162

¹Note that this AUC is a bit lower than the reported value of 81.2% in **Table 6.3**, because this value was obtained by taking the mean over 100 bootstrap samples, thus a slight change in reported value is expected.

6.6 Summary

These results are promising and should be investigated further with a larger dataset to establish statistical significance.

Chapter 7

Conclusion and Future Development

In this thesis we have investigated the use of cough audio data to aid in the diagnosis of tuberculosis (TB). To our best knowledge, this is the first system of its kind.

The design of the diagnosis system included the implementation of an automatic cough segmenter (annotator) to detect cough episodes within continuous recordings in a controlled environment. Hidden Markov models (HMMs) were used for this purpose. The HMM system was designed to detect three sound events (cough, silence and other), with a reported frame accuracy of 87.16%.

Various feature extraction methods were investigated. Filterbank energy based features were compared with Mel-frequency cepstral coefficient (MFCC) features. It was found that filterbank energy features outperformed MFCC features. This finding seems to indicate that the cough audio contains information relevant to TB diagnosis that is not perceivable by the human auditory system.

For the design of an audio-based classifier, two machine learning algorithms were evaluated: logistic regression and Gaussian mixture models (GMMs). Logistic regression outperformed the GMMs. The best reported area under the ROC curve (AUC) for logistic regression and GMM respectively were 81.2% and 79.2%.

The audio-based classifier results were compared to results achieved on objective clinical measurements for the same patients in the audio dataset. For classification of the clinical (meta) data, two classifiers were evaluated: logistic regression and decision trees. The logistic regression classifier outperformed the decision tree classifier with respective AUCs of 77.36% and 56.44%. Compared to the meta classifier, the audio-based classifier achieved a 3.84% higher AUC.

This finding indicates that the cough audio data contains more relevant information than the meta data for TB diagnosis.

By isolating different frequency bands, we investigated which parts of the frequency spec-

trum are most useful for cough classification. It was found that including the frequency band corresponding to vowel sounds (0-1105Hz) led to the best performance. Thereafter, we performed feature selection using a greedy search and a forward search algorithm. The forward search algorithm led to a system with an AUC improvement of 13.74% compared to a system trained on all features.

The best system after feature selection was then combined with the best meta data classifier using logistic regression as a fusion method. This led to a system with a sensitivity, specificity, accuracy, AUC and kappa of 82.35%, 80.95%, 81.58%, 94.34% and 0.6867 respectively.

In conclusion, this thesis serves as a proof of concept that TB diagnosis by cough sound analysis is possible. We hope that the promising results obtained can be taken further in future work.

7.1 Future Development

Should this project be developed further, the following aspects should receive attention.

Firstly, the dataset should be expanded. As a starting point, the number of TB negative patients could be increased, as this was a limiting factor in this study. Ethical clearance for the addition of new, healthy patients was acquired in the last stages of this study, but due to time constraints, this data could not be collected and pre-processed for inclusion into this study. The expansion of the dataset would hopefully reduce the variance and would thus enable more conclusions to be made regarding statistical significance of the findings in this study.

Data from patients with diseases similar to TB should also be included. The value of this system would be best utilized when developed to distinguish between borderline cases - when a patient is suspected of having TB but could be confused with having another pulmonary disease such as bronchitis or lung cancer.

Additional classifiers such as artificial neural networks (ANN), long short term memory (LSTM) neural networks or support vector machines (SVMs) could be investigated for cough classification. Neural networks model complex hypotheses by recursively learning different layers of weights applied to (usually non-linear) activation functions through a back propagation algorithm. For the use of neural networks a much larger database would be necessary. Support vector machines aim to separate samples by projecting the data into higher dimensions using kernel functions, which enable more complex class separation.

The addition of new information could also be investigated. Including clinical information such as HIV status (which was unavailable to us for all our patients) could benefit this

7.1 Future Development

system. Introducing data from different sources, such as forced expiratory volume over 1 second (FEV_1) could introduce insights into the physical status of the lung and to distinguish between asthma patients who could mistakenly be thought to have TB at a low cost.

In practice, a diagnosis system such as described in this study would run on an embedded platform such as a mobile phone. For this implementation, further research could be done into feature selection to lower the computational cost. Additionally, the integration of the automatic cough annotation system (HMM) and the cough classifiers could be implemented such that after a recording is taken, the cough episodes are isolated and classified in a seamless process.

In terms of reporting, a confidence indication (how confidently the system believes the patient to have TB) could be implemented. Ultimately a doctor would still make the final diagnosis, thus an indication of the probability that the patient has TB would enable the health practitioner taking the test to make a more informed decision, rather than just knowing the system's final classification results.

Although a bit far from current developments, something that might stem from this research is a patient recovery tracking system. When using this diagnosis tool, the recorded diagnostics could be captured, along with other patient information (name, address, next of kin, contact details, etc.), and be used to:

1. Keep relevant patient information in a digital profile to de-clutter and optimize working environments.
2. Schedule follow up examinations with patients, including reminders through SMS/email to patients and next of kin.
3. Track patient recovery over the course of treatment by comparing current system diagnosis confidence with previous results.
4. Expand the dataset with data gathered by active usage.

As mentioned in **Section 1.5.5**, ensuring completion of TB medication courses is key to making progress in the global battle against TB. Creating a system that helps track patient recovery, while aiding in the diagnosis process and at the same time optimizing the patient information control would add great value to our society.

Appendix A

Audacity Audio Editing Tool



Audacity is a free, open source audio recording and editing tool available at <http://www.audacityteam.org/>.

Audacity was used for manually annotating audio recordings.

The annotation process starts with loading a recording into Audacity's waveform display GUI, shown in **Figure A.1**. The automatic 'Sound Finder' tool is used to detect all sound events in the audio. A threshold value is chosen which determines which sounds are events and which are regarded as silence. Combined with the 'Sound Finder' tool, the 'Silence Finder' tool is used to detect all silence events, which labels all sounds below the chosen threshold with a 'S' label and is thus the inverse of the 'Sound Finder' tool. A threshold value of -35dB was chosen for both the sound and silence detection in the displayed figures. However, the threshold value was manually tuned for each file to give the best detection of cough sounds.

Figure A.2(a) shows a zoomed view of the recording in **A.1**. **Figure A.2(b)** shows the 'Silence Finder' and 'Sound Finder' tools displayed in the drop-down list of Audacity's Analyse menu. **Figure A.2(c)** shows the same zoomed recording after applying the Sound- and Silence Finder tools. Notice that the silence labels are only starting points. A script was later written to convert the labels shown in **A.2(c)** so that the silence labels extend between sound events. **Figure A.2(d)** shows the full recording in **A.1** with all detected sound and silence labels displayed.

After generating the sound and silence labels, the annotator would listen to the recording while scrolling through the labels. Each label would then be named as 'C' for cough, 'A' for ambient noise and so on. This process was repeated for each recording in \mathcal{D}_{AA} and \mathcal{D}_{CC} .

After each recording was annotated, the labels were exported to a text file, using the 'Export Labels' option (not shown in the figures). An example of such a label file is shown in **Figure A.2**. The values are arranged in the format (start, stop, label), with all times in seconds.

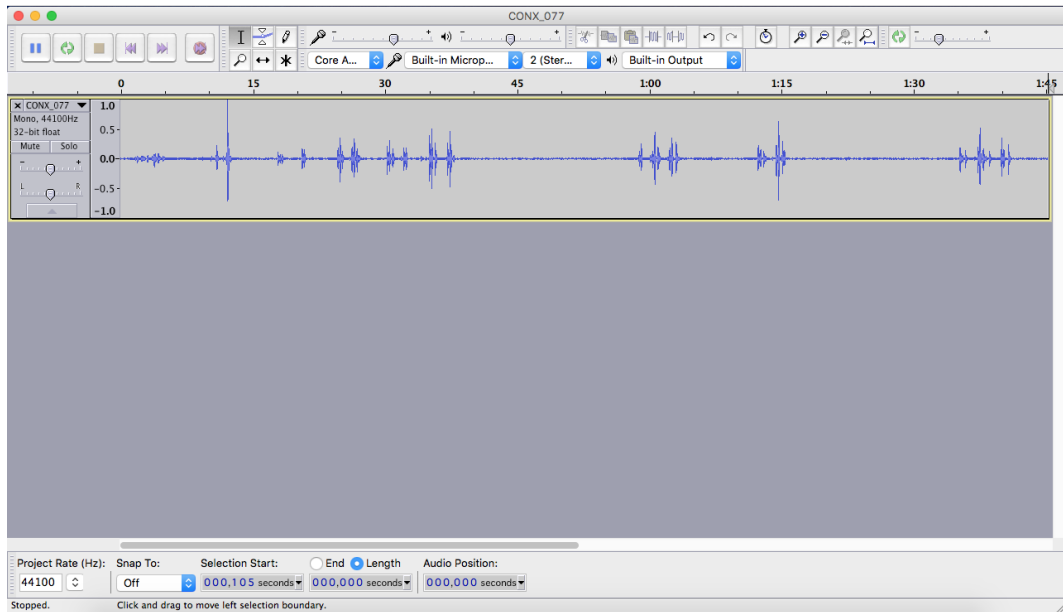
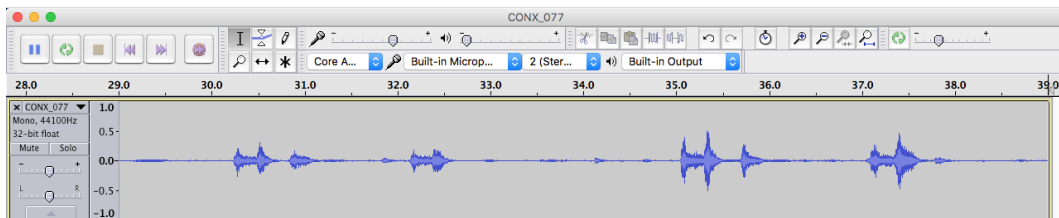
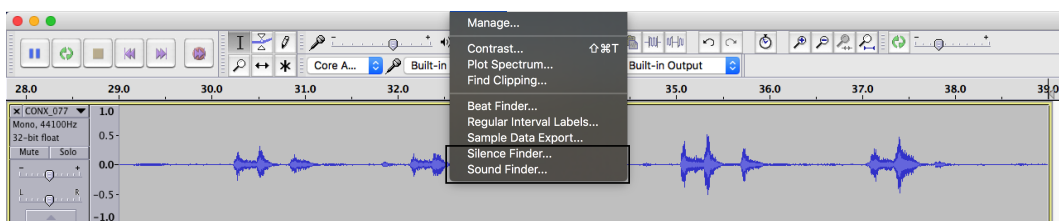


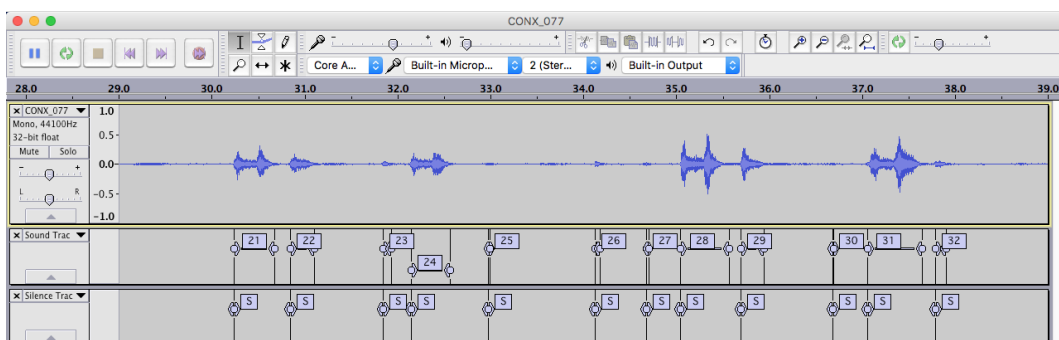
Figure A.1: Audacity GUI showing an already loaded recording.



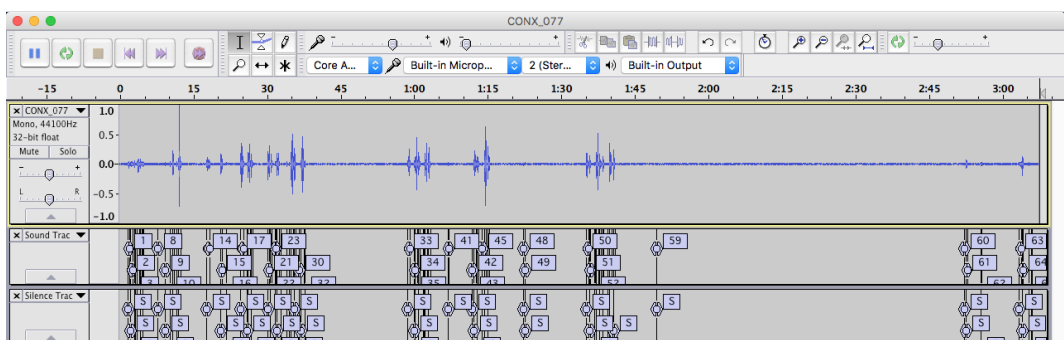
(a) Audio waveform zoomed to display 10 seconds of audio.



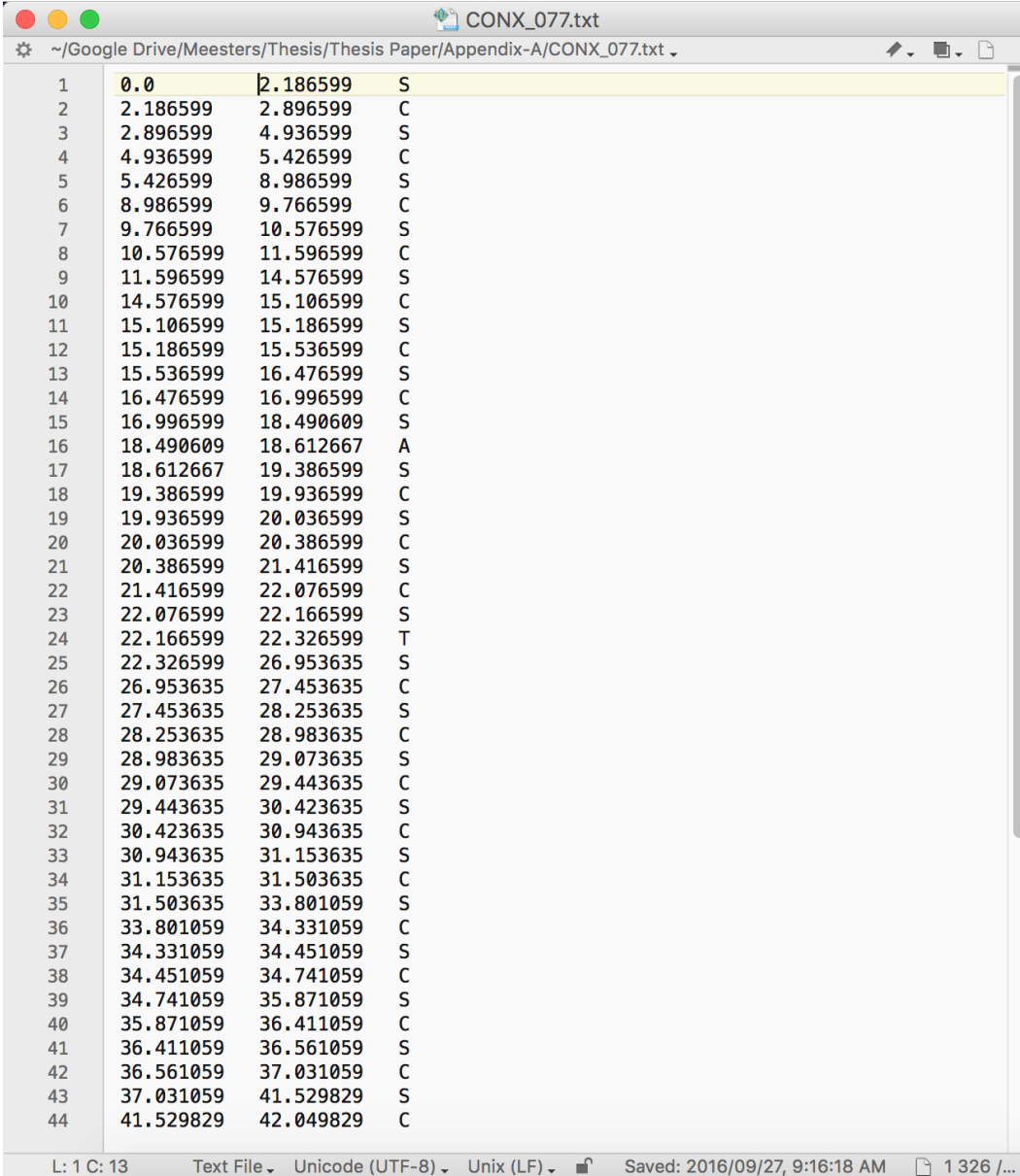
(b) Sound- and Silence Finder tools shown in the Audacity menu.



(c) Zoomed audio segment with labels after applying Sound- and Silence Finder tools. The top bottom bar contains the silence labels and the bar above that contains the event labels.



(d) Full waveform with sound and silence labels.



The image shows a text editor window titled "CONX_077.txt" with a file path of "~/Google Drive/Meesters/Thesis/Thesis Paper/Appendix-A/CONX_077.txt". The window displays a table with 44 rows. Each row contains four columns: a line number (1-44), a numerical value, a second numerical value, and a single-letter annotation (S, C, A, or T). The first row is highlighted in yellow.

Line	Value 1	Value 2	Annotation
1	0.0	2.186599	S
2	2.186599	2.896599	C
3	2.896599	4.936599	S
4	4.936599	5.426599	C
5	5.426599	8.986599	S
6	8.986599	9.766599	C
7	9.766599	10.576599	S
8	10.576599	11.596599	C
9	11.596599	14.576599	S
10	14.576599	15.106599	C
11	15.106599	15.186599	S
12	15.186599	15.536599	C
13	15.536599	16.476599	S
14	16.476599	16.996599	C
15	16.996599	18.490609	S
16	18.490609	18.612667	A
17	18.612667	19.386599	S
18	19.386599	19.936599	C
19	19.936599	20.036599	S
20	20.036599	20.386599	C
21	20.386599	21.416599	S
22	21.416599	22.076599	C
23	22.076599	22.166599	S
24	22.166599	22.326599	T
25	22.326599	26.953635	S
26	26.953635	27.453635	C
27	27.453635	28.253635	S
28	28.253635	28.983635	C
29	28.983635	29.073635	S
30	29.073635	29.443635	C
31	29.443635	30.423635	S
32	30.423635	30.943635	C
33	30.943635	31.153635	S
34	31.153635	31.503635	C
35	31.503635	33.801059	S
36	33.801059	34.331059	C
37	34.331059	34.451059	S
38	34.451059	34.741059	C
39	34.741059	35.871059	S
40	35.871059	36.411059	C
41	36.411059	36.561059	S
42	36.561059	37.031059	C
43	37.031059	41.529829	S
44	41.529829	42.049829	C

The status bar at the bottom indicates: L: 1 C: 13, Text File, Unicode (UTF-8), Unix (LF), Saved: 2016/09/27, 9:16:18 AM, 1 326 /...

Figure A.2: Example of output text file containing annotations.

References

- [1] N. A. Knechel, "Tuberculosis: Pathophysiology, Clinical Features, and Diagnosis," *Critical Care Nurse*, vol. 29, no. 2, pp. 34–43, 2009.
- [2] Y. Amrulloh, U. Abeyratne, V. Swarnkar, and R. Triasih, "Cough Sound Analysis for Pneumonia and Asthma Classification in Pediatric Population," *Proceedings - International Conference on Intelligent Systems, Modelling and Simulation (ISMS)*, vol. 6, pp. 127–131, 2015.
- [3] P. S. Hershey, "Respiratory System Information," Web, July 2016. [Online]. Available: <http://pennstatehershey.adam.com/content.aspx?productId=117&pid=2&gid=9248>
- [4] T. F. Institute, "Body Systems: Respiratory System - The Human Heart: An Online Exploration from The Franklin Institute, Made Possible by Unisys," Web, May 2013. [Online]. Available: <http://learn.fi.edu/learn/heart/systems/respiration.html>
- [5] WebMD, "Human Respiratory System and Lungs; How They Work," Web, September 2014. [Online]. Available: <http://www.webmd.com/lung/how-we-breathe>
- [6] A. Suresh. (2015, July) What Really Happens During a Cough? Web. [Online]. Available: <http://www.thehealthsite.com/diseases-conditions/understanding-the-cough-process/>
- [7] V. Swarnkar, U. R. Abeyratne, Y. a. Amrulloh, and A. Chang, "Automated Algorithm for Wet/Dry Cough Sounds Classification," *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 3147–3150, Aug 2012.
- [8] H. Chatrzarrin, A. Arcelus, R. Goubran, and F. Knoefel, "Feature Extraction for the Differentiation of Dry and Wet Cough Sounds," *2011 IEEE International Symposium on Medical Measurements and Applications*, pp. 162–166, May 2011.
- [9] T. Y. Corp. (2016) The Human Auditory System. [Online]. Available: http://www.yamahaproaudio.com/global/en/training_support/selftraining/audio_quality/chapter4/01_ear_anatomy/

-
- [10] B. Tempel, “Human Auditory System,” Handout (Web). [Online]. Available: https://depts.washington.edu/tempelab/06NuB502_{_}handout.doc
- [11] “Fluid Wave - How Hearing Works — HowStuffWorks.” [Online]. Available: <http://health.howstuffworks.com/mental-health/human-nature/perception/hearing4.htm>
- [12] “The Cochlea : an alternate to Fourier transform.” [Online]. Available: <https://crilfdvinternshipromoli.wordpress.com/2015/12/23/the-cochlea-an-alternate-to-fourier-transform/>
- [13] S. Errede, “The Human Ear Hearing , Sound Intensity and Loudness Levels,” Lecture Notes (University of Illinois - Physics 406), pp. 1–34, 2002.
- [14] Monty, “24/192 Music Downloads are Very Silly Indeed,” Web, March 2012. [Online]. Available: <http://people.xiph.org/~xiphmont/demo/neil-young.html>
- [15] Behzad Munir, “Voice Fundamentals – Human Speech Frequency,” Web, 2012. [Online]. Available: <http://www.uoverip.com/voice-fundamentals-human-speech-frequency/>
- [16] Prof. Jeffrey Hass, “How do we perceive pitch?” Handout (An Acoustic Primer, Indiana University), 2003. [Online]. Available: <http://www.indiana.edu/~emusic/acoustics/pitch.htm>
- [17] World Health Organization, “Tuberculosis Report 2015,” Report, pp. 50–51, 2015. [Online]. Available: http://www.who.int/tb/publications/global_report/gtbr15_main_text.pdf
- [18] World Health Organization , “Tuberculosis Fact Sheet,” Web, Oct 2016. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs104/en/>
- [19] A. Kanabus. Information about tuberculosis - statistics. [Online]. Available: [InformationaboutTuberculosis](http://www.who.int/mediacentre/factsheets/fs104/en/)
- [20] “TB Statistics for South Africa — National & provincial,” 2016. [Online]. Available: <http://www.tbfacts.org/tb-statistics-south-africa/>
- [21] South African Department of Health, “Annual performance plan 2012/13 - 2014/2015,” Tech. Rep., 2012. [Online]. Available: <http://www.tbfacts.org/wp-content/uploads/2015/06/App2012-2014.pdf>
- [22] South African National Aids Council , “National Strategic Plan on HIV, STIs and TB 2012-2016,” Tech. Rep., 2011. [Online]. Available: <http://sanac.org.za/wp-content/uploads/2015/11/National-Strategic-Plan-on-HIV-STIs-and-TB.pdf>

-
- [23] National Institution of Allergy and Infectious Diseases. Tuberculosis, A Detailed Explanation. [Online]. Available: <http://www.niaid.nih.gov/topics/tuberculosis/Understanding/WhatIsTB/pages/detailed.aspx>
- [24] No Author, "Tuberculosis - What Happens." [Online]. Available: <http://www.webmd.com/lung/tc/tuberculosis-tb-what-happens>
- [25] C. M. Muvunyi, F. Masaisa, and Pintelaan, "Diagnosis of Smear-Negative Pulmonary Tuberculosis in Low-Income Countries : Current Evidence in Sub-Saharan Africa with Special Focus on HIV Infection or AIDS." *Understanding Tuberculosis - Global Experiences and Innovative Approaches to the Diagnosis*, pp. 128–146, 2012. [Online]. Available: <http://www.intechopen.com>
- [26] A. Kanabus. Information about Tuberculosis. [Online]. Available: <http://www.tbfacts.org/tb-tests/>
- [27] World Health Organization, "Use of tuberculosis release assays (IGRAs) in low- and middle- income countries," 2011.
- [28] S. Nayak and B. Acharjya, "Mantoux Test and Its Interpretation." *Indian Dermatology Online Journal*, vol. 3, pp. 2–6, Jan 2012. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3481914/>
- [29] C. C. Boehme, P. Nabeta, D. Hillemann, M. P. Nicol, S. Shenai, F. Krapp, J. Allen, R. Tahirli, R. Blakemore, R. Rustomjee, A. Milovic, M. Jones, S. M. O'Brien, D. H. Persing, S. Ruesch-Gerdes, E. Gotuzzo, C. Rodrigues, D. Alland, and M. D. Perkins, "Rapid molecular detection of tuberculosis and rifampin resistance," *New England Journal of Medicine*, vol. 363, no. 11, pp. 1005–1015, 2010, PMID: 20825313. [Online]. Available: <http://dx.doi.org/10.1056/NEJMoa0907847>
- [30] World Health Organization,, "Xpert MTB/RIF Test," 2013. [Online]. Available: <http://who.int/tb/laboratory/mtbrifrollout>
- [31] P. Piirila and A. R. A. Sovijarvi, "Differences in Acoustic and Dynamic Characteristics of Spontaneous Cough in Pulmonary Diseases," *Chest*, pp. 46–53, 1989.
- [32] C. R. Woolf and A. Rosenberg, "Objective Assessment of Cough Suppressants Under Clinical Conditions Using a Tape Recorder System." *Thorax*, vol. 19, pp. 125–130, 1964.
- [33] J. Martinek, P. Klco, M. Vrabec, T. Zatko, M. Tatar, and M. Javorka, "Cough sound analysis," *Acta Medica Martiniana*, vol. 1, 2013.

-
- [34] S. S. Birring, T. Fleming, S. Matos, a. a. Raj, D. H. Evans, and I. D. Pavord, "The Leicester Cough Monitor: Preliminary Validation of an Automated Cough Detection System in Chronic Cough." *The European Respiratory Journal*, vol. 31, no. 5, pp. 1013–8, May 2008. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/18184683>
- [35] B. H. Tracey, G. Comina, S. Larson, M. Bravard, J. W. López, and H. Robert, "Cough Detection Algorithm for Monitoring Patient Recovery from Pulmonary Tuberculosis," *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6017–6020, 2011.
- [36] T. Drugman, J. Urbain, N. Bauwens, R. Chessini, A.-s. Aubriot, P. Lebecque, and T. Dutoit, "Audio and Contact Microphones for Cough Detection," *Interspeech*, pp. 1–4, 2012.
- [37] J. Korpa and J. Sadlon, "Methods of Assessing Cough and Antitussives in Man - Analysis of the Cough Sound," *Pulmonary Pharmacology*, vol. 9, pp. 261–268, 1996.
- [38] U. R. Abeyratne, V. Swarnkar, A. Setyati, and R. Triasih, "Cough Sound Analysis Can Rapidly Diagnose Childhood Pneumonia." *Annals of Biomedical Engineering*, vol. 41, no. 11, pp. 2448–62, Nov 2013. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/23743558>
- [39] W. Thorpe, M. Kurver, G. King, and C. Salome, "Acoustic analysis of cough," *Australian and New Zealand Intelligent Information Systems Conference*, vol. 7, pp. 18–21, November 2001.
- [40] A. Murata, Y. Taniguchi, Y. Hashimoto, Y. Kaneko, Y. Takasaki, and S. Kudoh, "Discrimination of productive and non-productive cough by sound analysis," *Internal Medicine*, vol. 37, no. 9, pp. 5–8, 1998.
- [41] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. John Wiley & Sons, 2001.
- [42] D. Ramage, "Hidden Markov models fundamentals," Lecture Notes (Stanford CS229), pp. 1–13, 2007. [Online]. Available: <http://see.stanford.edu/materials/aimlcs229/cs229-hmm.pdf>
- [43] R. J. Elliot, L. Aggoun, and J. B. Moore, *Hidden Markov Models*. Springer Science And Business Media, 1995.
- [44] J. Eisner, "An Interactive Spreadsheet for Teaching the Forward-Backward Algorithm," *Proceedings of the ACL02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational*

- Linguistics*, vol. 1, pp. 10–18, July 2002. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1118108.1118110>
- [45] S. Raschka. (2016) Regularization in Logistic Regression: Better Fit and Better Generalization? Web. [Online]. Available: <http://www.kdnuggets.com/2016/06/regularization-logistic-regression.html>
- [46] R. Sridharan, “Gaussian Mixture Models and the EM Algorithm Review : the Gaussian Distribution,” Lecture Notes (MIT), pp. 1–11.
- [47] E. M. Thomas, A. Temko, G. Lightbody, W. P. Marnane, and G. B. Boylan, “Gaussian Mixture Models for Classification of Neonatal Seizures Using EEG,” *Physiological Measurement*, vol. 31, no. 7, pp. 1047–1064, 2010.
- [48] S. Alsmadi and Y. P. Kahya, “Design of a DSP-based Instrument for Real-Time Classification of Pulmonary Sounds.” *Computers in Biology and Medicine*, vol. 38, no. 1, pp. 53–61, Jan 2008. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/17716642>
- [49] K. Hajian-Tilaki, “Receiver Operating Characteristic (ROC) Curve Analysis for Medical Diagnostic Test Evaluation,” *Caspian J Intern Med*, vol. 4, no. 2, pp. 627–635, 2013. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/24009950>
- [50] T. G. Tape, “Using the Receiver Operating Characteristic (ROC) curve to analyze a classification model,” *University of Nebraska Medical Center*, pp. 1–3, 2000. [Online]. Available: [http://www.math.utah.edu/~gamez/files/ROC-Curves.pdf\\$delimiter"026E30F\\$nhhttp://gim.unmc.edu/dxtests/roc2.htm](http://www.math.utah.edu/~gamez/files/ROC-Curves.pdf$delimiter)
- [51] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 6, pp. 299–309, 2005.
- [52] A. Harries, D. Maher, and S. Graham, “TB/HIV- A clinical manual.” World Health Organization, Tech. Rep. 2, 2004. [Online]. Available: <http://www.who.int/maternal{ }child{ }adolescent/documents/9241546344/en/>
- [53] C. Wejse, P. Gustafson, J. Nielsen, V. F. Gomes, P. Aaby, P. L. Andersen, and M. Sodemann, “TBscore: Signs and Symptoms from Tuberculosis Patients in a Low-Resource Setting Have Predictive Value and May Be Used to Assess Clinical Course.” *Scandinavian Journal of Infectious Diseases*, vol. 40, no. 2, pp. 111–120, 2008.
- [54] S. Young, G. Evermann, M. Gales, D. Hain, Thomas Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge University Engineering Department, 2009.

-
- [55] S. Davis and P. Mermelstein, “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences,” *IEEE Transactions on Acoustic, Speech and Signal Processing*, no. 4, 1980.
- [56] G. Strang, “The Discrete Cosine Transform.” [Online]. Available: www-math.mit.edu/~gs/papers/dct.pdf
- [57] M. Westphal, “The Use of Cepstral Means in Conversational Speech Recognition.” *Proceedings of the European Conference on Speech Communication and Technology (Eurospeech)*, pp. 1143—1146, 1997. [Online]. Available: <http://20.210-193-52.unknown.qala.com.sg/archive/archive{-}papers/eurospeech{-}1997/e97{-}1143.pdf>
- [58] V. Vrabie, P. Granjon, and C. Serviere, “Spectral Kurtosis: From Definition to Application,” *IEEE International Workshop on Nonlinear Signal and Image Processing (NSIP)*, vol. 6, pp. 8–10, 2003.
- [59] F. Moreno-Seco, J. Iñesta, P. D. León, and L. Micó, “Comparison of Classifier Fusion Methods for Classification in Pattern Recognition Tasks,” *Lecture Notes in Computer Science*, pp. 705–713, 2006. [Online]. Available: <http://www.springerlink.com/index/G6338U8831M32826.pdf>
- [60] No Author. (2002) Understanding Audiograms, Hearing Loss, and Speech Intelligibility. [Online]. Available: <https://www.hdhearing.com/learning/part2.htm>
- [61] H. Bisani M, Ney, “Bootstrap Estimates for Confidence Intervals in ASR Performance Evaluation,” *Proceeds of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, 1988.
- [62] I. Guyon and A. Elisseeff, “An Introduction to Variable and Feature Selection,” *Journal of Machine Learning Research (JMLR)*, vol. 3, no. 3, pp. 1157–1182, 2003.