Sirindhorn International Institute of Technology

Thammasat University

Thesis ICTES-MS-2009-03

AUTOMATIC HEART SOUND ANALYSIS FOR TELE CARDIAC AUSCULTATION

Sumeth Yuenyong

AUTOMATIC HEART SOUND ANALYSIS FOR TELE CARDIAC AUSCULTATION

A Thesis Presented

by

Sumeth Yuenyong

Master of Engineering
Information and Communication Technology for Embedded Systems
(ICTES) Program
Sirindhorn International Institute of Technology

Thammasat University
May 2010

AUTOMATIC HEART SOUND ANALYSIS FOR TELE CARDIAC AUSCULTATION

A Thesis Presented

By

Sumeth Yuenyong

Submitted to

Sirindhorn International Institute of Technology

Thammasat University

In partial fulfillment of the requirement for the degree of

MASTER OF ENGINEERING

Approved as to style and content by

Advisor and Chairperson of Thesis Committee	Assoc. Prof. Waree Kongprawechnon, Ph.D.
Committee Member and Chairperson of Examination Committee	Kanokvate Tungpimonrut, Ph.D.
Committee Member	Prof. Akinori Nishihara, Ph.D.
Committee Member	Itthisek Nilkhamhang, Ph.D.

External Examiner : Assoc. Prof. Sanya Mitaim, Ph.D.

May 2010

Acknowledgement

The author wishes to express his gratitude to his advisor and co-advisor, Asst. Prof. Dr. Waree Kongprawechnon and Prof. Dr. Akinori Nishihara, for their guidance, encouragement and insightful comments throughout my whole research progress.

Special thanks to the Industrial Control and Automation (ICA) laboratory of National Electronic and Computer Technology Centre (NECTEC) for providing the electronic stethoscope and to the doctors of the Thammasat University (TU) hospital's heart center for their valuable comments.

And finally, thank you to my parents for the support and care they have always given me.

This research is supported by Thailand Advanced Institute of Science and Technology - Tokyo Institute of Technology (TAIST-Tokyo Tech), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology (Tokyo Tech) and Sirindhorn International Institute of Technology (SIIT), Thammasat University (TU).

The author and his advisor received financial support by a scholarship from Telecommunication Research and Industrial Development Institute (TRIDI) under grant No.Master/001/2551 and is hereby acknowledged.

Abstract

Automatic Heart Sound Analysis for Tele Cardiac Auscultation

May 2010

by

Sumeth Yuenyong

B.Eng, Sirindhorn International Institute of Technology

Thammasat University, 2004

This thesis is concerned with the algorithm development for an automatic heart sound analysis system that is a part of a tele-auscultation project. The goal of tele-auscultation is to create a device that can automatically acquire and analyse the heart sound of a patient in order to determine if it shows any sign of cardiac disorder or not. This project is motivated by the apparent imbalance of cardiac medical capability between hospitals in Bangkok and those in other areas of Thailand.

Heart sound analysis process consists of segmentation, feature extraction and classification. Segmentation is the most difficult process due to heart sounds are non-stationary signal and interference from extra cardiac sounds. This research presents a novel method that does not require the segmentation of heart sound into individual cardiac cycles. It also proposed a classification method based on using multiple neural networks acting as committee.

The method consists of envelope detection using the energy of wavelet transform coefficients and calculation of the cardiac cycle length using the autocorrelation of the envelope signal. Features extraction is performed on a five cardiac cycle segment of both the envelope signal and the heart sound signal itself. The raw feature space is normalized and then optimized using principal component analysis technique. The classifier is a committee of feedforward neural networks with ten members. Each of the ten neural networks was trained with training data sampled from the original feature space in a bootstrap process. Ten fold cross validation for each of the neural network showed average accuracy of 94 $\%$. The final performance of the classifier are 94.7 $\%$ accuracy, 93.3 $\%$ sensitivity and 100 $\%$ specificity.

Table of Contents

List of Figures

List of Tables

List of Symbols

$\mathbf{x}$: A feature vector
y : Output of an ANN
d : Desired output of an ANN
N : Total number of feature vectors
E : Envelope signal of a heart sound
T : Test set during a fold of cross-validation
$\mathbf{C}$: Covariance matrix
$\mathbf{Y}$: Matrix of wavelet coefficients
$\mathbf{X}$: Training set of feature vectors

List of Algorithms

Chapter 1

Introduction

This chapter describes the organization of this thesis, research motivation, system overview, and concludes with an introduction to the cardiac cycle and heart sounds.

1.1 Proposal Organization

This thesis is organized into six chapters. Chapter 2 provides a background on heart sound analysis. Chapter 3 reviews the related literatures of each specific stage of heart sound analysis. Chapter 4 describes the methodology that this research used to solve the problem. Chapter 5 presents the results and discussion, and finally Chapter 6 is the conclusion.

1.2 Research Motivation

Heart disease is a major health problem throughout the world. In Thailand, heart disease is responsible for about 35% of all deaths. Figure 1.1 shows a breakdown of 100000 deaths into different causes.

It can be seen that heart disease was the third leading cause of death. Furthermore, the distribution of doctors are significantly imbalanced between Bangkok and other areas of the country. Specialized doctors are situated in Bangkok area, while doctors in the rural areas tend to be general practitioner. This means that people who lives in the rural areas have almost no access to cardiologist. Even those who do have access may not get regular check up because government hospitals are very crowded and check up appointments can take as long as six months [1]. For these reasons, often cardiac disorders are not detected before symptoms had developed and the diseased had already advanced.

Cardiac disorders that are related to heart valves malfunction or abnormal heart rates can however, can be detected without the need for sophisticated equipments. Specifically, by listening carefully to the sound of the heart using a device called a *stethoscope*, an experienced cardiologist can detect the presence or even diagnose the type of the underlying cardiac disorder even in its early stage [2]. This diagnosis technique is known as *auscultation*, it requires minimal equipment, just a stethoscope, and it is also non-invasive and very

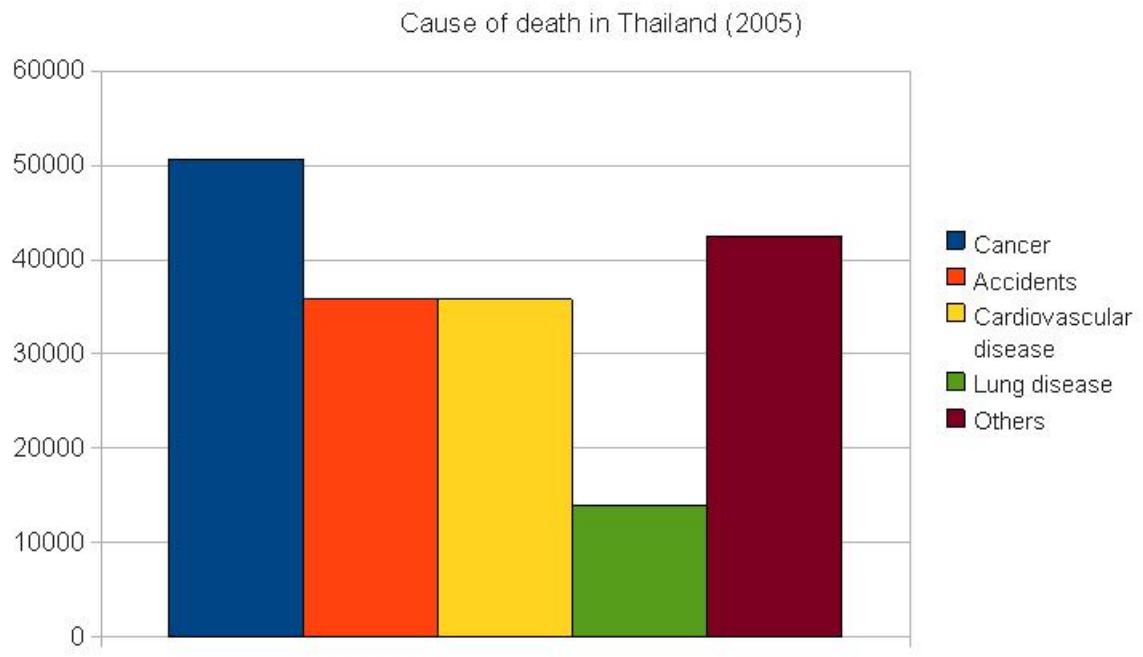


Figure 1.1: Causes of deaths in Thailand in 2005

cheap compared to other types of cardiac examination methods. These advantages make auscultation the most practical mean of cardiac examination in small primary health care facilities or clinics. This is especially true for rural areas in developing countries such as Thailand. Figure 1.2 illustrates auscultation being performed.

Even thought it is simple, performing auscultation effectively requires training and experience. This is because the key to auscultation involves detecting sounds that are low-pitch and short duration, which is difficult for the human hearing system [2]. Auscultation teaching and training had also been on the decline due to the availability of new high-technology diagnosis tools which are expensive and therefore only available in large hospitals[3]. These two factors had resulted in the decline of auscultation skills of younger doctors.

Motivated by these problems, an automatic tele-auscultation system is proposed. The system is to perform auscultation in place of a doctor. That is, it must acquire heart sound from a person and analyze it to detect the presence of a cardiac disease. The system should be able to perform these tasks automatically and it also has telecommunication capability so that the patient and doctor who are not at the same location can communicate. The goal is to provide an effective and low cost preliminary cardiac check up that can be deployed in small health clinics throughout the country in large numbers. A person who had been tested positive on this system can then be referred to a cardiologist, which can save time and money by reducing the number of unnecessary referrals. Ultimately, this can help reduce the number of deaths caused by heart disease by early detection.

Figure 1.2: Performing auscultation (http://meded.ucsd.edu/clinicalmed/heart.htm)

1.3 System Overview

The overview of the system is depicted in Figure 1.3 and Figure 1.4

The proposed tele-auscultation system is an example of tele-medicine. In the framework of tele-medicine as shown in Figure 1.3, the patient, medical personnel (not necessarily a doctor but anyone trained to use the equipment) and diagnosis station are at different physical locations and are linked together through the Internet. In this work, the concern is on the diagnosis station, which in this case will be at the same physical location as the patient. The diagnosis station is shown in Figure 1.4, it consists of a stereo camera, a robot arm, an electronic stethoscope mounted onto the end of the robot arm, and a standard PC. When a patient sits down in front of the examination table, the stereo camera detects the person and locates the auscultation site on his/her chest. The robot arm then moves the chest-piece of the stethoscope to the auscultation site and the stethoscope records the heart sound. An electronic stethoscope functions in exactly the same way as a regular stethoscope used by most doctors, with the addition that it can connect directly to a PC's soundcard. This way, the heart sound can be read into the PC, where a software performs analysis on the sound to form a diagnosis. The PC can also link to a doctor or other medical personnel through the Internet, who can communicate with the patient via video conference and also view the diagnosis result in real time. This research of this thesis focuses on developing the algorithm of the heart sound analysis software on the PC. Heart sound analysis means to determine

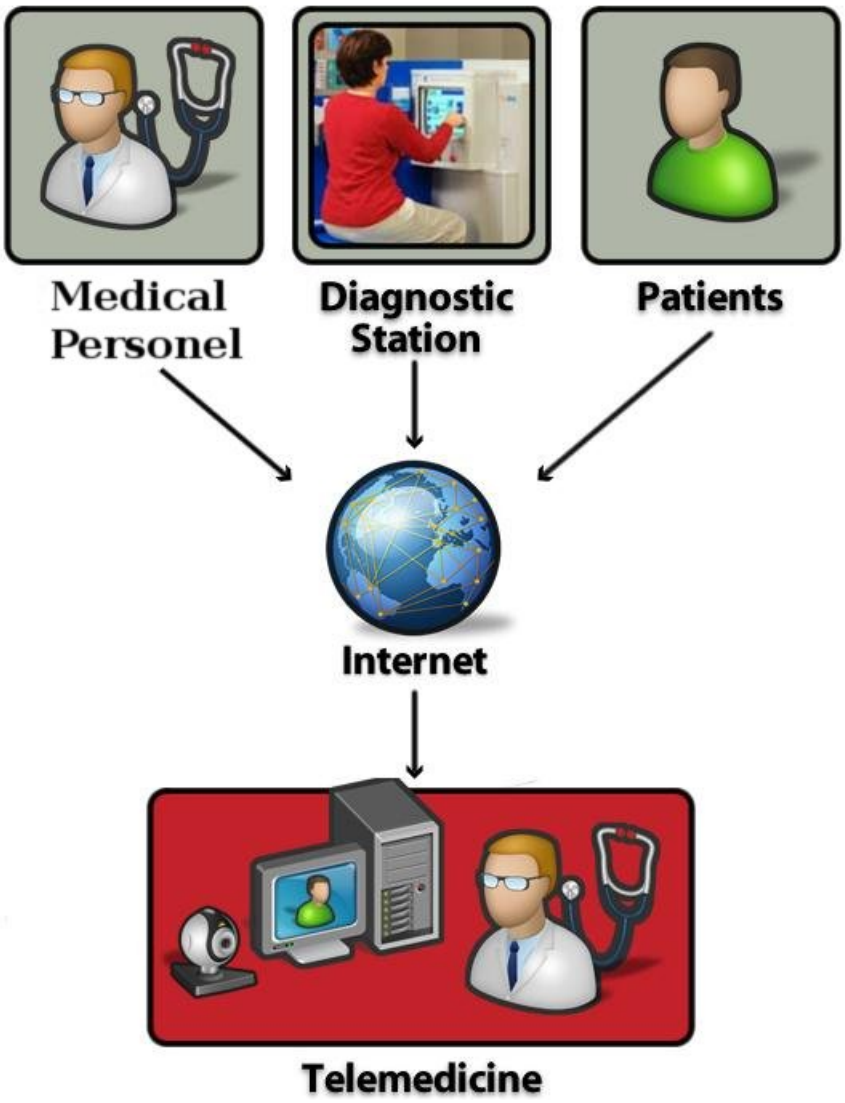


Figure 1.3: Telemedicine

whether or not the heart sound in question represents a healthy heart or a diseased one. If the result shown by the analysis software is positive (disease detected) then the patient is referred to more examinations for confirmation and more detailed diagnosis.

1.4 Heart, Heart Sounds, and Electrocardiogram (ECG)

This section provide the introduction to the heart, normal and abnormal heart sounds, and electrocardiogram (ECG) signal. The structure of the human heart is shown in schematic drawing in Figure 1.5. Medical diagrams always assume that the subject on the drawing is the patient that's facing the viewer, thus left and right sides are reversed.

The heart has four chambers, the top two chambers are called the Atriums and the bottom two are called the Ventricles. It has four valves: Aortic, Mitral, Pulmonary and

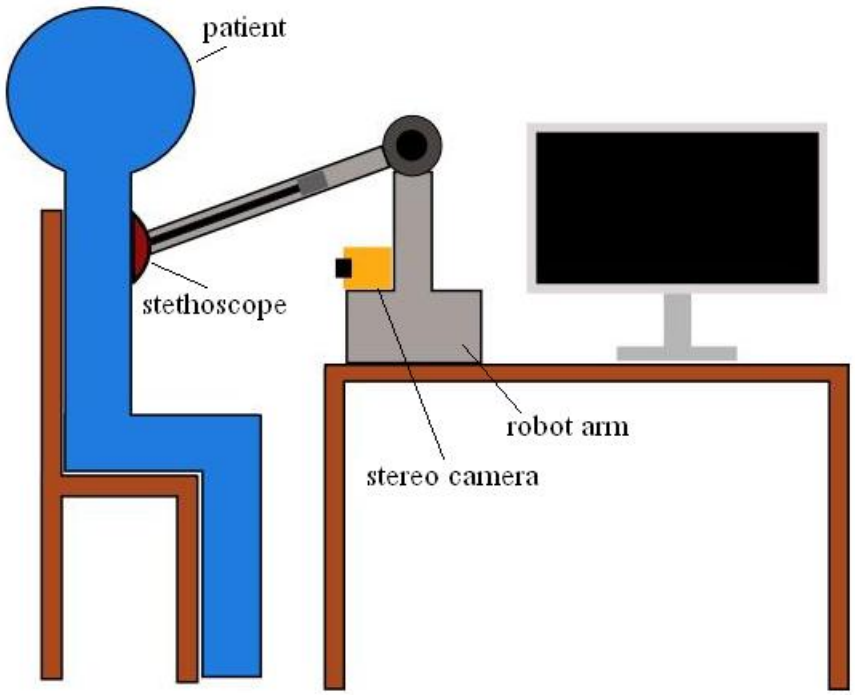


Figure 1.4: Overview of the system's operation

Tricuspid as shown by the trapezoids in Figure 1.5. They are referred to as A, M, P and T valve. The T and M valves connect the right and left chambers of the heart, respectively, and the P and A valves connect the ventricles to the two main arteries; the Aorta that goes to the rest of the body and the Pulmonary that goes to the lungs. The function of these valves is to keep blood flowing in the right direction as the heart contracts.

1.4.1 The Cardiac Cycle

A cardiac cycle represents a single heartbeat. Every single beat of the heart proceeds as follows; first, the two atriums (the top two chambers) are filled with blood returning from the body and from the lungs. The atriums then contract, the Tricuspid and Mitral valves open and blood is pushed into the ventricles. When the ventricles are filled, the Tricuspid and Mitral valves close to prevent blood from flowing back into the atriums. Then ventricles then begins to contract, increasing the pressure, once the pressure is high enough, the Aortic and Pulmonic valve open and blood is pushed into the arteries. When the ventricles are empty, the Aortic and Pulmonic valves are closed, preventing blood from flowing back into the ventricles as the pressure inside drops. The heart then relaxes in preparation for the next cycle.

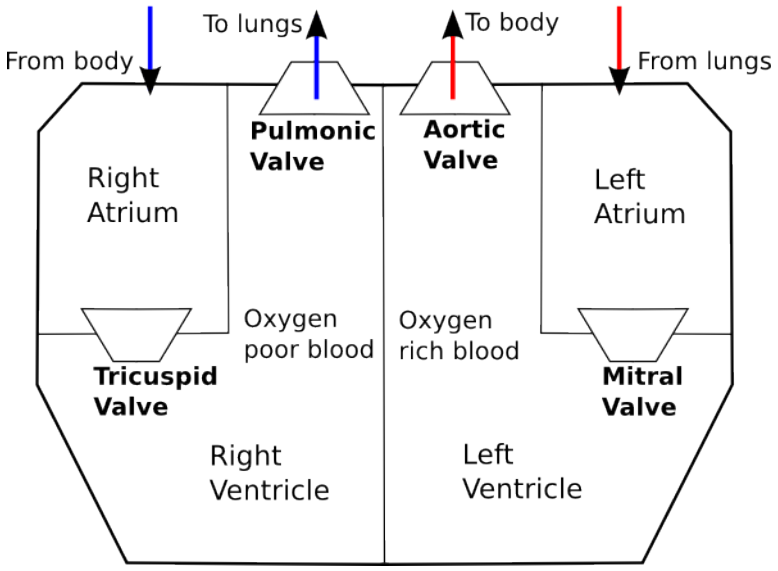


Figure 1.5: Schematic drawing of the heart

1.4.2 Heart Sounds

Heart sounds are a result of vibrations due to the closure of heart valves. In normal adults there are two heart sounds; the first heart sound (S1) and the second heart sound (S2). Together, they are referred to as *fundamental heart sound* (FHS). The S1 sound is due to the closure of the T and M valves and S2 is due to the closure of the A and V valves. Using these heart sound as markers, the cardiac systole is defined as the period between the end of S1 and the beginning of S2, while the cardiac diastole is between the end of S2 and beginning of the next cycle's S1. Figure 1.6 shows one cardiac cycle of heart sounds.

1.4.3 Abnormal Heart Sounds

Abnormal heart sounds contain components that are not one of the FHS. These extra components fall into two types, cardiac sounds and extracardiac sounds. Sounds of the first type that are strongly correlated to cardiac disorders are the third heart sound (S3) and the fourth heart sound (S4). Figure 1.7 shows one cardiac cycle with S3 which is located in early diastole and Figure 1.8 shows a cardiac cycle with S4. The presence of S3 or S4 in adults are considered pathologic [4]. It is assumed in this research that heart sound analysis will be performed on adult patients, thus heart sound with S3 or S4 are always considered abnormal.

Extra cardiac sounds are caused by turbulence in blood flow due to high flow rate through a blocked valve, or backward flow through an leaking valve. Heart murmurs are extra sounds that are not part of a healthy cardiac cycle, they can be heard either during systole or diastole, or both. They are caused by turbulent blood flow due to one or more heart valves

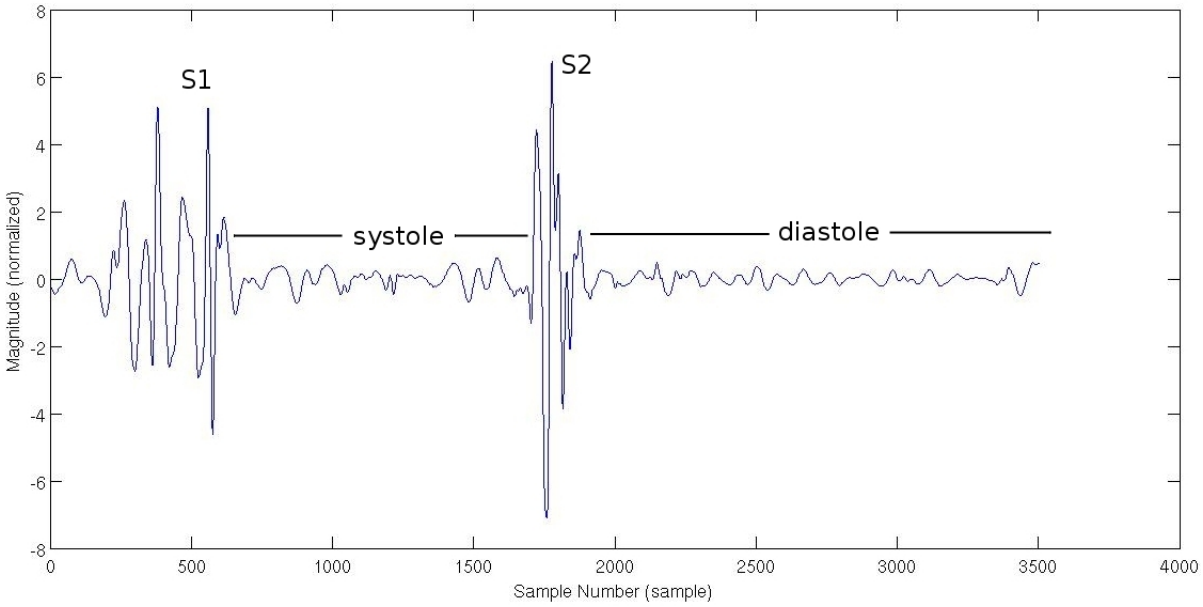


Figure 1.6: A normal cardiac cycle

leaking (regurgitation), or blocked (stenosis). As a result, their presence is a good indicator of valvular disorders. In fact, the detection and classification of extra components in a cardiac cycle that are not the FHS is the key to cardiac auscultation. There are many different types of murmur, each with different magnitude and frequency characteristic. Murmurs also range from very soft in early cases to very loud in advanced cases. Figure 1.9 shows a cardiac cycle with a type of murmur called aortic regurgitation which is caused by leakage of the aortic valve.

It can be seen that abnormal heart sounds are characterized by extra components in their cardiac cycles that are not FHS. Therefore, in order to recognize abnormal heart sounds, it is necessary to first determine the boundaries of cardiac cycles, which involves identifying all the S1 and S2 in a heart sound signal since a cardiac cycle is defined as the period between consecutive S1 sounds. This process is called *segmentation*, it can be a difficult process for certain types of heart sounds and often an extra signal called the electrocardiogram (ECG) signal is used as a marker to perform segmentation.

1.4.4 ECG Signal

An ECG signal is the recording of the electrical activity inside the heart over time using electrodes. This signal is the electrical pulse inside the heart that instructs the heart muscle when to contract. The plot of ECG signal time-correlated with heart sound is shown in Figure 1.10.

The ECG signal display impulse-like characteristics. Each part of the ECG cycle is

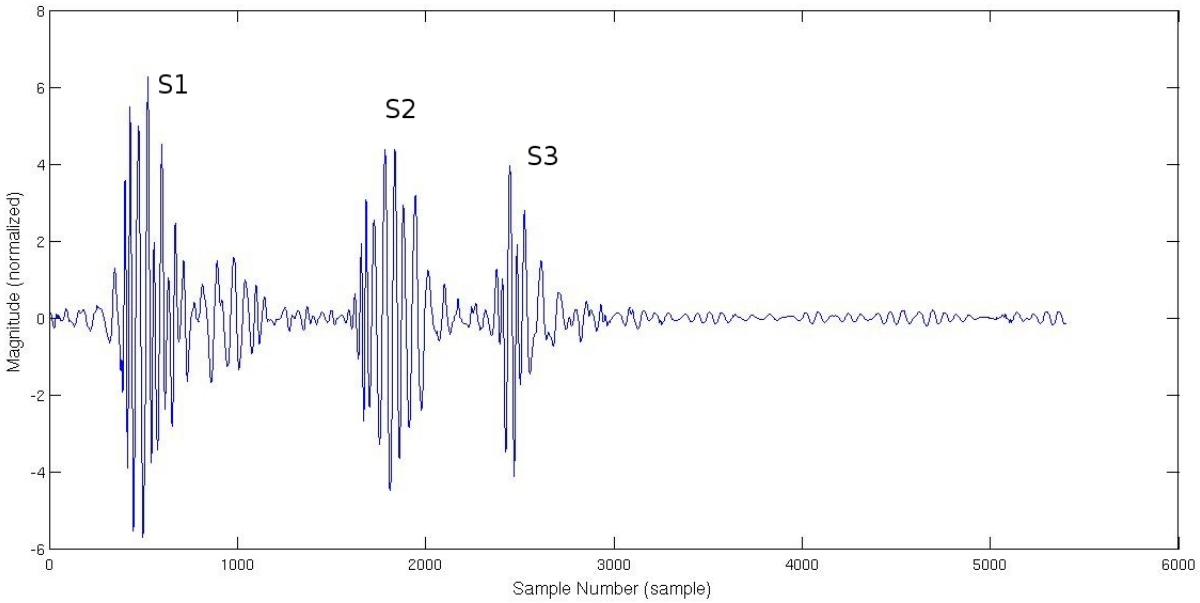


Figure 1.7: A cardiac cycle with third heart sound

referred to as ”wave” followed by a letter that indicate its position. Thus, the spikes in the ECG signal are called the QRS complex for they are formed by the Q, R and S wave. The heart sound and ECG has a specific relationship. The S1 sound comes immediately after the R wave and the S2 comes no later than 150 ms after the T wave on the ECG signal. This relationship between the ECG and heart sound had been utilized in many studies such as [3] to perform segmentation of heart sound. Incorporating the ECG makes segmentation simpler than working with the heart sound alone, because the locations of QRS can easily detect because of their impulse-like characteristic. This in turn allows for simple locating of S1 sounds which mark the beginning of cardiac cycles. Using the ECG signal however requires extra equipments to record the ECG signal which increases the cost and complexity of the system.

1.5 Research Goals

The goal of this research is to devise an algorithm that can distinguish between normal and abnormal heart sounds. While this may sound modest because no identification of a particular cardiac disease is attempted, this study incorporates more different types of heart sounds than many researches reported in the literature. In particular, many studies focus on detection and classification of just the four main type of heart murmur; aortic regurgitation/stenosis, and mital regurgitation/stenosis. However, there are many types of abnormal heart sounds that are not murmur; third heart sound, fourth heart sounds, opening snap, systolic click, split S2 and summation gallop. These sounds are strong indicators of cardiac diseases but are rarely considered in heart sound analysis literature. They are also harder to

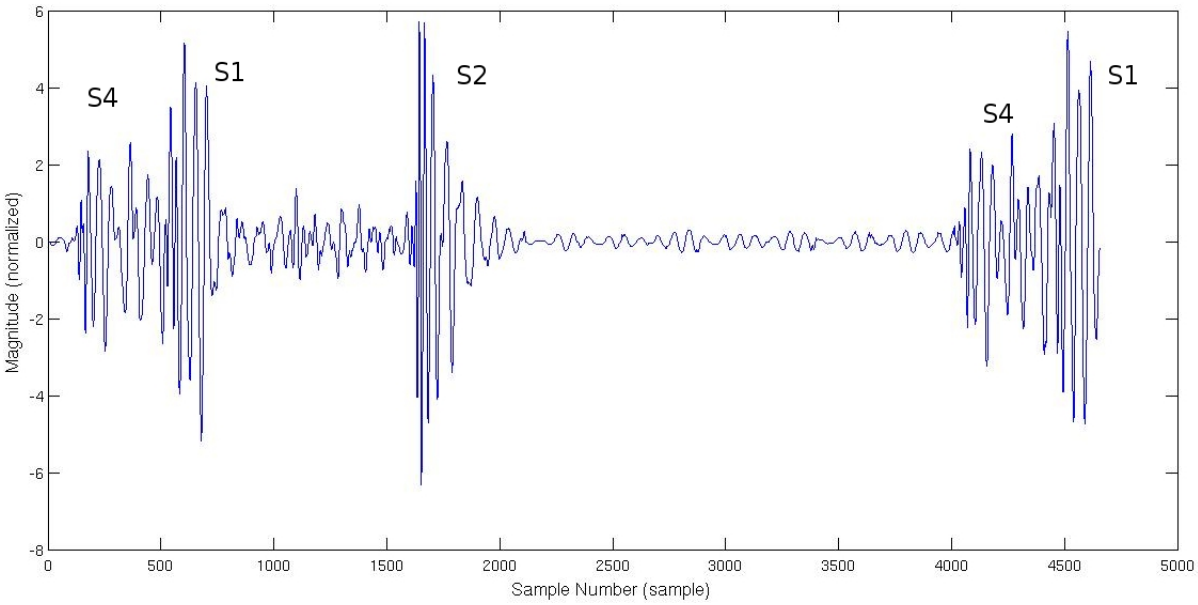


Figure 1.8: A cardiac cycle with fourth heart sound

detect than murmur because they tend to have lower intensity and shorter duration. This research attempts to fill this gap by providing an algorithm that can detect all types of abnormal heart sounds

The algorithm also does not rely on any axillary signal like the ECG. This is keeping with the goal that the final automatic auscultation system should simple and low-cost. Even thought Using ECG signal simplifies heart sound analysis by providing precise timing information, it also means extra equipment is need to record the ECG and the electrodes must be manually placed on the patient, thus the system no longer fully automatic.

This chapter had explained the motivation for this research, which is seriousness of heart disease, how people in the rural area have no access to cardiologist, and some heart diseases can be detected early by auscultation performed by a skillful doctor. A low cost system that can perform auscultation in place of a doctor would be a great advantage in detecting some heart diseases early on a large scale. This research deals with the algorithm behind heart sound analysis, which is the core of this system. The goal is to be able to detect all types of abnormal heart sounds.

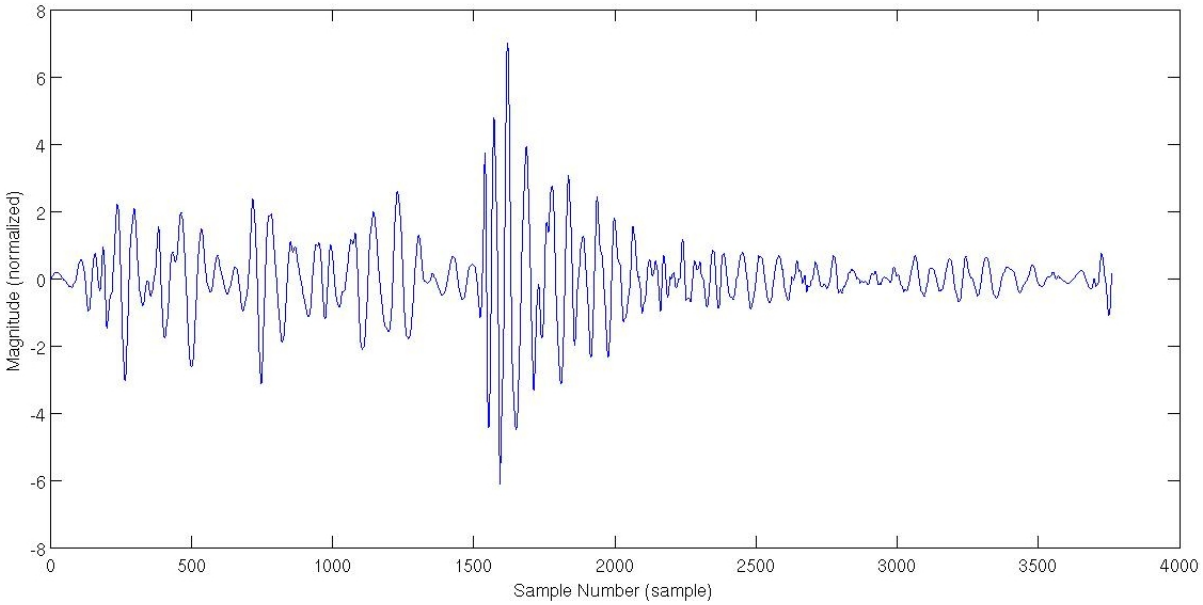


Figure 1.9: A cardiac cycle with aortic regurgitation

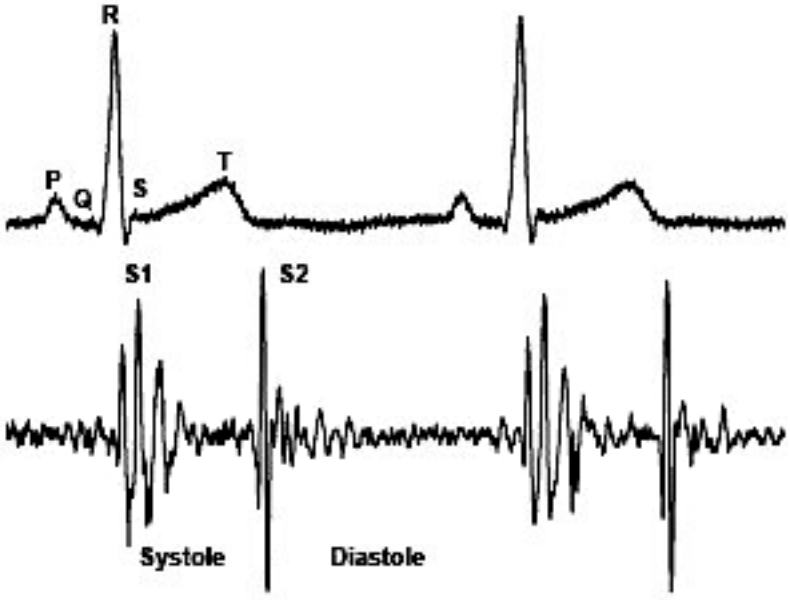


Figure 1.10: Heart sound and ECG time-correlated

Chapter 2

Background Information

This chapter provides background information on heart sound analysis. As described in Chapter 1, the goal of heart sound analysis is essentially to detect extra components within a cardiac cycle. Unfortunately, heart sounds are complicated signals that can vary significantly from person to person, for this reason, heart sound analysis is based on statistical pattern recognition approach.

2.1 Statistical Pattern Recognition

Statistical pattern recognition deals with classification of objects. An object is any entity of interest that one would like to classify, for example, a person's face in facial recognition application, or a heart sound in the case of this study. Classification means to assign a class label to an object, such as identification of good and defective products in automatic visual inspection, one may want to assign a class label of zero for good product and a class label of one for a defective product. The goal of statistical pattern recognition is to assign correct class labels to a group of objects. The entity that assign a class label to an object is called a classifier. Classifiers are evaluated in terms of their performance, which comprises of three numerical criterion. First is the accuracy, which is ratio of correctly classified samples over all samples. Second is sensitivity, which is the ratio of correctly classified positive (which in this work means diseased heart sounds) samples over all positive samples. The final criteria is the specificity, which is the ratio of correctly classified negative (healthy heart sounds) samples over all negative samples. When general "performance" refers to accuracy, since it is the more general performance criteria.

2.1.1 Features

In any application of statistical pattern recognition, a set of characteristics or traits of the objects to be classified must first be chosen. The characteristics can be anything that provide sufficient identification information in order to classify an object. These characteristics are represented by real numbers for the case of measurable quantities such as length or weight, and by integers in the case of attributes such as colors. Each of these numbers

is referred to as *features*. A set of features representing the characteristics of an object are concatenated together to form a *feature vector*. A feature vector can be thought of as an numerical identification tag that is attached to an object. The process of calculating feature vectors for objects is known as *feature extraction*.

Features are selected based on the problem under consideration and there is no general procedure that can come up with an appropriate set of features for any problem. Feature selection is thus based on experience of the problem under consideration and trial and error. It may seem intuitive at first that the more features are available, the better the classification accuracy. However, having too many features can actually decrease the accuracy, this is called ”the curse of dimensionality”, in which the problem becomes more and more complicated as more features are added and the recognition accuracy suffers. The concept of the curse of dimensionality can be illustrated with a classic example of an image recognition system that differentiate between the alphabet ”a” and ”b” where each image is represented by a 128 by 128 matrix of grayscale values. In the extreme case, a feature vector can be formed by concatenating each row (or column) of the matrix into single 16384 elements vector. It can immediately be seen that using such large number of features just to tell the difference between ”a” and ”b” is not feasible. On the other hand, one could use just a single feature, the ratio between the character's height and width. This ratio tends to be larger for ”b” since it is taller than for ”a”. This will most likely give better performance than using 16384 features. This could be improved by adding a second feature, the size of the area enclosed by the character, which also tend to be larger for ”b” than for ”a”. The addition of the second feature will increase the recognition accuracy further but not as much as the introduction of the first feature. Adding more and more features result in smaller improvement in accuracy until a certain number whereby adding more features decreases the performance. Classification performance increases with increasing number features until a certain point, then decrease as more feature are added. Thus one would seek to come up with the feature set that maximizes the classification performance for the problem at hand. This is the problem of feature selection.

2.1.2 Feature Selection

It is common practice in statistical pattern recognition to first have as many features variable as one is able to come up with. Evaluating different subsets of these features and selecting only some of these features to actually use. It can be shown that the optimum feature set given an initial set can only be obtained through exhaustive search. This is impractical for most problems, thus suboptimal methods must be employed. There are many algorithm available to come up with a ”good” feature set. In this work, principal component analysis (PCA) was used [5]. The advantage of PCA is that in addition to reducing the number of features, the remaining features are linearly combined such that a new feature variable is obtained from projecting all the data points onto an axis along which the variance is maximum.

The variance of a feature variable has a direct implication on performance of the classifier based on them, this is discussed in the next section while PCA is discussed in details in chapter 4.

2.2 Naive Bayesian Classifier

Naive Bayesian classifier is the simplest classifier regime in which one assign a class label that maximizes the posterior probability of an object being a that class given the observation (feature vector) Consider a simple problem with two classes with a single feature. One assign an object with feature value x to class c_i such that $p(x|c_i)$ is maximized. Naive Bayesian classifier assumes that for each class of objects to be recognized, a feature is a random variable with Gaussian pdf, as shown in Figure 2.1

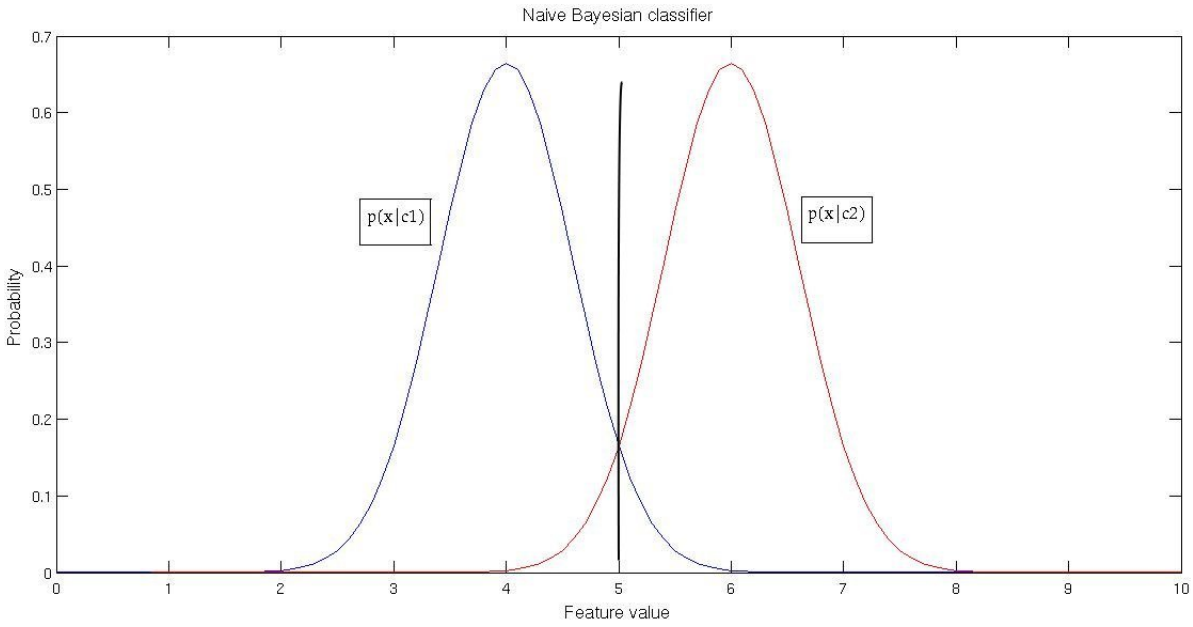


Figure 2.1: Naive Bayesian classification

In Figure 2.1 the horizontal axis is the feature value and the vertical axis is the probability of observing each feature value. It can be seen that there are two separate pdf's, the one on the left is the probability of observing x given that it is from class 1, $p_1(x|c_1)$ and the one on the right is for class 2, $p_2(x|c_2)$. Given an observation, classification is simply look up the probabilities from the two pdf curves and assign the object to the class with higher probability. This is based on the well-known Bayes's theorem:

$$p(c_i|x) = \frac{p(c_i)p(x|c_i)}{p(x)}, \tag{2.1}$$

where $p(c_i|x)$ is called the posterior probability (probability after making an observation) of class c_i given the observation x, $p(c_I)$ is called the prior (the probability of c_i before making an observation) and $p(x|c_i)$ is the likelihood of x given that it came from class c_i. The numerator of the right hand side of Equation 2.1 and $p(c_i)$ in the numerator are constant given a data set. Thus to maximize the left hand side of Equation 2.1, the problem is to simply select class c_i such that $p(x|c_i)$ is maximized.

The vertical line in Figure 2.1 indicates the point where the two pdf curves cross, and hence the likelihood of both classes are the same at this point, $p_1(x|c_1)$ is higher to the left of this line and $p_2(x|c_2)$ is higher to the right. Therefore, the feature value where the two pdf cross can be used as a threshold value, if an observation is below the threshold, then class 1 is assigned otherwise class 2 is assigned. This is the principle of naive Bayesian classification.

In most practical applications however, no single feature is enough to provide reliable classification because the pdf curves may not be far apart enough for naive Bayesian classification to work accurately, as illustrated in Figure 2.2. The two pdf curve are not sufficiently far apart such that using naive Bayesian approach will result in high miss-classification rate because even though the likelihood may be higher for one class, it is still significantly high for another class. This shows that the variance of a feature variable directly affect the performance of the classifier, the larger the variance of a feature, the easier it is to classify based on this feature. In this sense the PCA method discussed in chapter 4 is a very good feature optimizing procedure since it maximizes the variance of each feature variable.

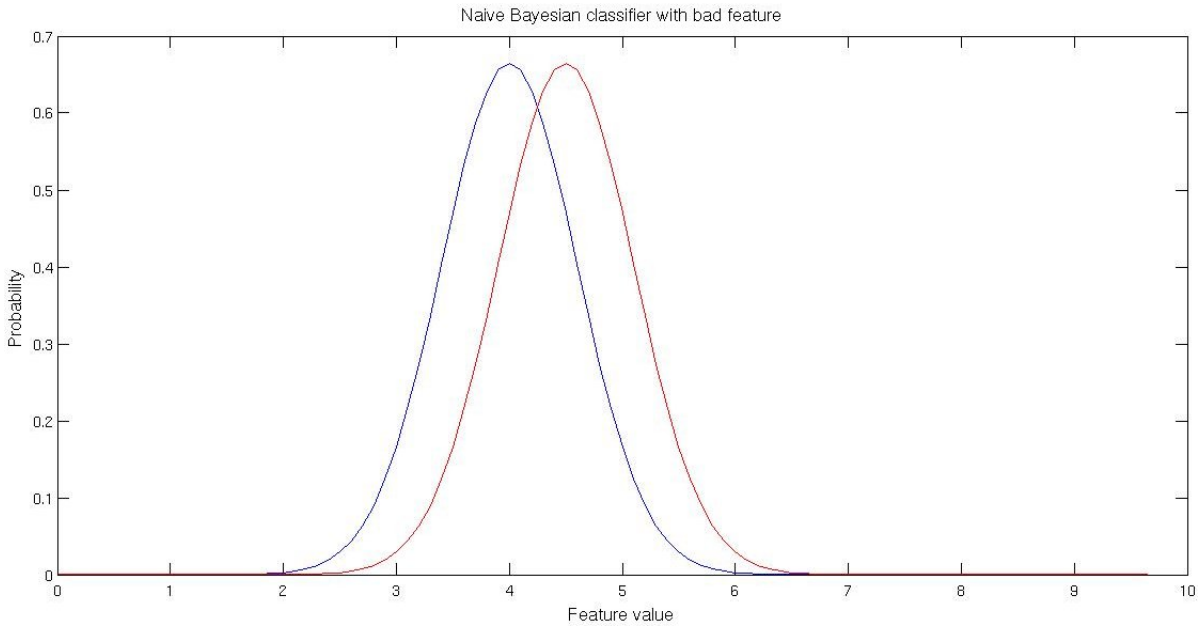


Figure 2.2: Naive Bayesian classification with bad feature variable

Because in practice no single feature variable will result in pdf curves that are nicely

separated like those in Figure 2.1. It is therefore necessary to use more than one features in the form of feature vectors. Using multiple features makes the Gaussian distributions become multi-dimensional and it is impossible to visualize high-dimensional pdf's. Naive Bayesian classification handles the multi-dimensional case by assuming that each feature variables are independent from each other so each can be treated as one dimensional Gaussian pdf and makes decision according to the following

$$i = \underset{c_i}{\operatorname{argmax}} \prod_{n=1}^{N} p(x_n|c_i), \tag{2.2}$$

where i is the class label, x_n is a feature variable and N is the total number of features. Naive Bayesian classifier is simple and quite effective. However, it has a few disadvantages. First, the independent assumption is often not true and secondly quite large training data is needed to accurately estimate the mean and variance of the pdf's.

2.3 Model Based Classifiers

Naive Bayesian classifiers are simple and gives reasonable performance. However more sophisticated classifiers are available that outperform the basic naive Bayesian. They are based on building a mathematical model that takes a feature vector as input and generates a class label as output. The model is trained by the training set of objects to be recognized, which must contain multiple objects of each class, and should be large enough to account for the within-class variance of the objects. The training set must be labeled, i.e., the correct class of each object must be known. Feature extraction is performed on the training set to generate a set of feature vectors that represents the training set. Each n-dimension vector corresponds to a point in n-dimensional space. A linear space populated with points specified by the feature vectors is called a *feature space*.

In the training phase, the classifier partitions the input space into mutually exclusive regions, where each region contains only the feature points that correspond to objects of the same class, this is illustrated in Figure 2.3 for the case two-dimensional feature space. The blue crosses are objects in one class and the green circle are objects in another.

Figure 2.3 shows the case where the number of classes is two. The red line divides the feature space into two regions, objects that fall on one side of the line is said to be in one class and those the fall on the other side belongs to the other class. The boundary is the right hand side of Equation 2.3 which is a line in two dimensions or a hyperplane in high dimensions. The vector $\mathbf{w}$ is the vector of weights or coefficients defining the hyperplane.

$$y(\mathbf{x}) = \mathbf{w^T x} + \mathbf{w_0} \tag{2.3}$$

It can be seen from Figure 2.3 that no single straight line can provide a perfect classification, some crosses inevitably fall on the side of the circles and some circles fall on the side of

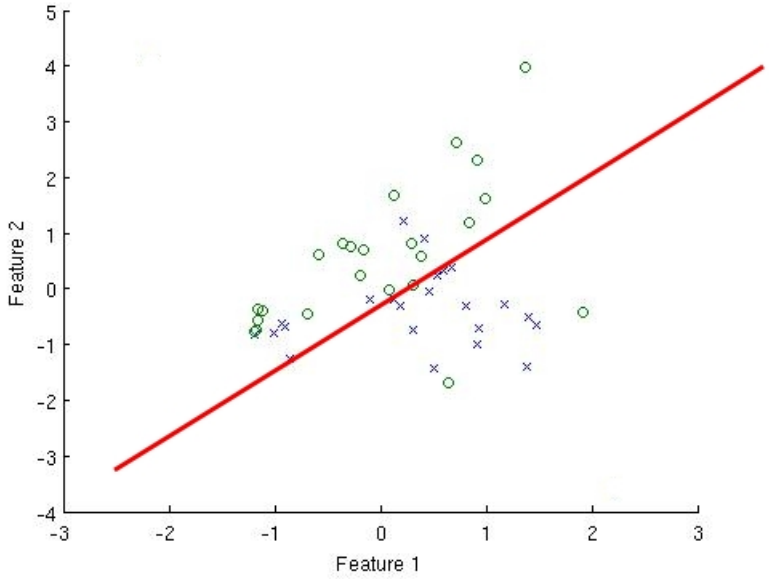


Figure 2.3: A partitioned 2D feature space with linear boundary

the crosses. In this case the feature space is said to be not linearly separable. The goal of classifier training is to adjust w so that the resulting region boundary yield the smallest number of misclassification

During training, the training set is input into the classifier and the parameters of the classifier are adjusted in order to minimize some objective function that provides a measure of error between the desired output (the known class label) and actual output of the classifier. In the case of classifier based on Equation 2.3, the measure of error could be the minimum distant to the boundary. Once training is completed, the parameters of the classifier is fixed. An unknown feature vector for an object that is not in the training set can be classified by inputting it into the classifier and noting class label outputted by the classifier.

A certain type of mathematical model called artificial neural network (ANN) is a popular choice for classifier for statistical pattern recognition in general and in heart sound analysis in particular. This is due to the non-linear nature of ANN that allows them to non-linearly partition the feature space, i.e., the boundaries of the regions can have arbitrary shape, as apposed from straight lines in the linear case. ANN is quite commonly used in heart sound analysis [6],[7],[8],[9].

2.4 Artificial Neural Network

An ANN is a network of individual computing elements called neurons. It is represented by a graph as shown in Figure 2.4

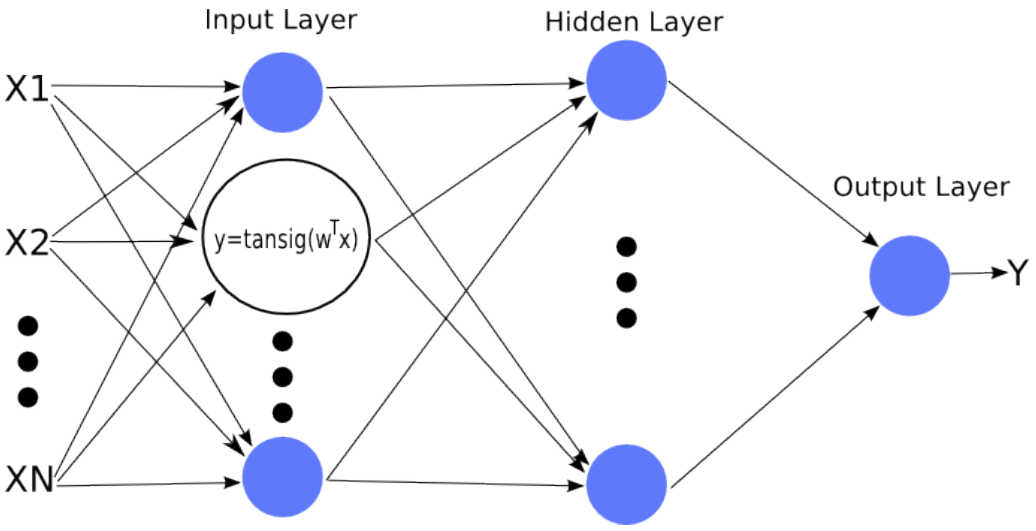


Figure 2.4: A simple ANN

In Figure 2.4, each node represents an individual computing element called neurons while each line represents a connection between a pair of neurons. The direction of the arrows indicate the direction of the connection. ANN can be organized into different network topologies, the one seen in Figure 2.4 where the connection are exclusively from left to right (all the arrows are pointing to the right) is called the feedforward topology, which is used extensively in statistical pattern recognition. Under this topology, neurons are organized into layers, in a graphical representation, neurons within the same layer are aligned vertically as can be seen in Figure 2.4. Also in feedforward topology there are three layers in total, the leftmost layer is called the input layer, the second layer is called the hidden layer and the rightmost layer is called the output layer. The big neuron in Figure 2.4 indicates the operation inside of each neuron discussed in the next section.

2.4.1 Neurons

Neurons are simple nonlinear computing elements. A neuron has multiple inputs and a single output, each input has an associated *weight*. The weights of neurons are the free parameters of an ANN and they are adjusted during training. A neuron's computation is simple; the inputs are multiplied with their corresponding weights and added to together, the sum is called a *activation*. The activation becomes the input of an *activation function*, the output of which is the output of the neuron. The big neuron in Figure 2.5 shows an Equation for the computation inside a neuron with tan-sigmoid activation function. The computation of a neurons follow Equation 2.4, where $\mathbf{w}$ is a vector of a neurons' connection weights, $\mathbf{x}$ is a vector representing the input and f is an activation function [10].

$$y = \text{tansig}(\mathbf{w}^{\mathbf{T}}\mathbf{x}) \tag{2.4}$$

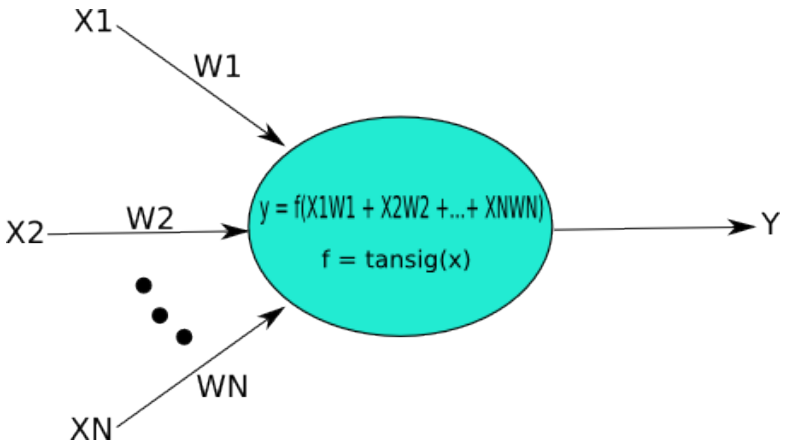


Figure 2.5: Inside a neuron

There are many type of activation functions, regardless of the type, an activation function maps the input domain of $(-\infty\ \infty)$ to output with range $[-1\ 1]$. One particular activation function, the *tan sigmoid* function, is a popular choice for an activation because it is differentiable, which is a desirable property when training an ANN. The formula of tan-sigmoid function is given in Equation 2.5 where the parameter a determine the shape of the curve, a group of plots with different value of a is shown in Figure 2.6.

$$y = \frac{2}{1 + \exp(-2ax)} - 1 \tag{2.5}$$

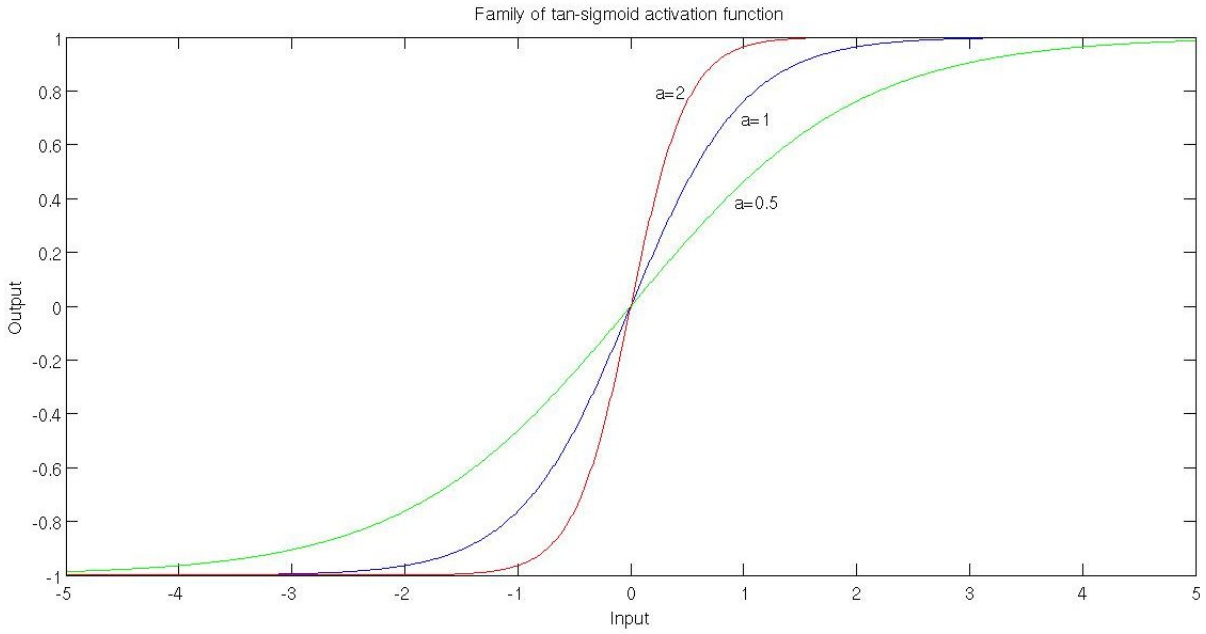
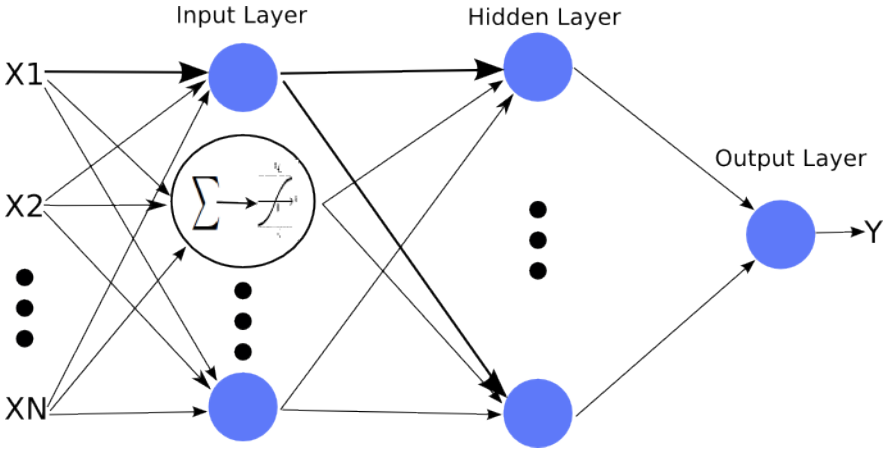


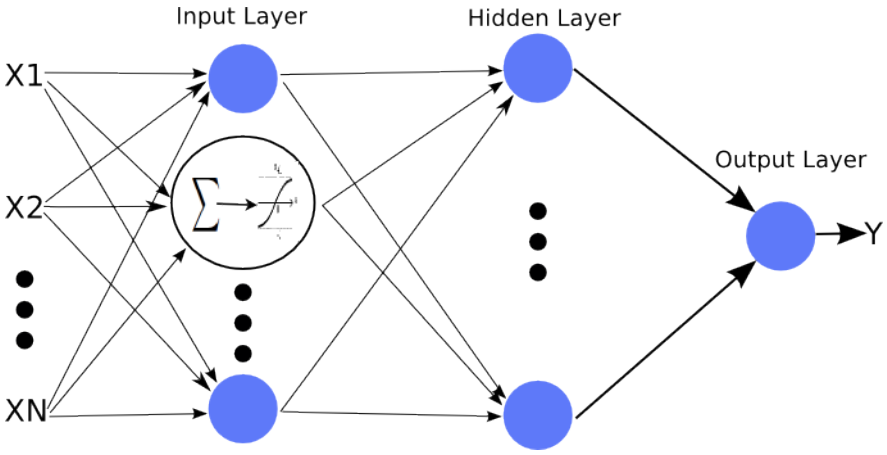
Figure 2.6: Tan-sigmoid activation function

2.4.2 Signal Flow in ANN

A feedforward ANN computes its output by propagating the input signal from left to right. Each neuron in the input layer receive the input vector $\mathbf{x} = \mathbf{x_1}, \mathbf{x_2}, ...\mathbf{x_n}$ and compute their outputs according to Equation 2.4. The output of each neuron in the input layer is distributed to each and every neuron in the hidden layer. This is shown in Figure 2.7 for the top neuron in the hidden layer where the thick connections indicate that a signal is flowing through it. Each neurons in the hidden layer then compute their outputs which feed into the output layer and the output of the output layer is the output of the ANN itself. The signal flow shown in Figure 2.7 is due to the top neuron in the input layer. In reality, each neuron would also perform this computation and contribute to the calculation of the ANN's output.



(a) Signal flow from input to hidden layer



(b) Signal flow from hidden layer to output

Figure 2.7: Computing ANN output

2.4.3 Setting Number of Neurons

In feedforward ANN used in statistical pattern recognition, the number of neurons in the input layer is fixed to be the same as the dimension of the input vector. The number of neurons in the output layer is dependent on the number of classes to be classified. For a two classes problem, the number of output neurons is just one, and its output is ideally zero (or -1) for a class and 1 for another class. If the number of classes is more than two, the number of output neurons is equal to the number of classes. Each neuron in the output layer corresponds to one class, where the output of a neuron is one while all the other neurons are zero when the input vector is of the same class. This means that the number neurons in the input and output layers of feedforward ANN are fixed by the dimension of the input vector and the number of classes. The number of hidden neurons however, are free to be chosen to maximize the performance of the ANN.

Since each element of the input vector is connected to every neurons in the input layers, whose numbers are fixed to be the same as that of the length of the input vector. Therefore for N elements feature vector there are a total of N^2 connections between the input vector the neurons in the input layer. For the output layer, in the case of two classes problem where there is only a single output neuron the number of connections between the input layer to the output layer is the same as the number of hidden neurons in the hidden layer. The number of neurons in the hidden layer can be set freely and the output of each neuron in the input layer go to every neuron in the hidden layer. Thus there are MN connections between the input and the hidden layer, where N is the number of neurons in the input layer and M is the number neurons in the hidden layer.

Each connection in an ANN has an associated connection weight, which are the free parameters of the ANN that get adjusted during training. The number of free parameters is very important in using ANN, in fact, it is very important to any statistical model. If the number of free parameter is not enough, there are not enough to be adjusted and the ANN will perform poorly. On the other hand, too many free parameter may result in *overfitting*, which is what happens when there are to many free parameters and not enough training data.

2.4.4 Overfitting

Overfitting occurs when there are too many free parameters. That is, the model is too complicated for the amount of training data available. For illustration, several plot showing the result of fitting the function $y = 5sin(x)$ with added normally distributed noise using ANN is shown in Figure 2.8. The number of datapoint is 100 and hidden neurons are 2, 5, 8 and 20. The circles indicated the training datapoints and the green line is the resulting fit. It can be seen that when hidden neurons is 2, the ANN performs poorly and the resulting curve does not resemble the sine wave at all. When hidden neurons is 5, the resulting curve

closely resembles the sine wave. When hidden neurons is increased to 8, the resulting curve still resemble the sine curve but with more error. Finally when hidden neurons is 20, the resulting curve goes back to poor performance again. In Figure 2.9, ANN with 20 hidden neuron was used to fit the curve again but this time the number of datapoints was increased to 1001, it can be seen that when the number of datapoints increases, the performance also increase given a fixed ANN. This shows that the number of hidden neuron must be carefully chosen given the problem at hand and the amount of training data available to obtain good curve fitting performance and the same principle also applies to using ANN for classification as well.

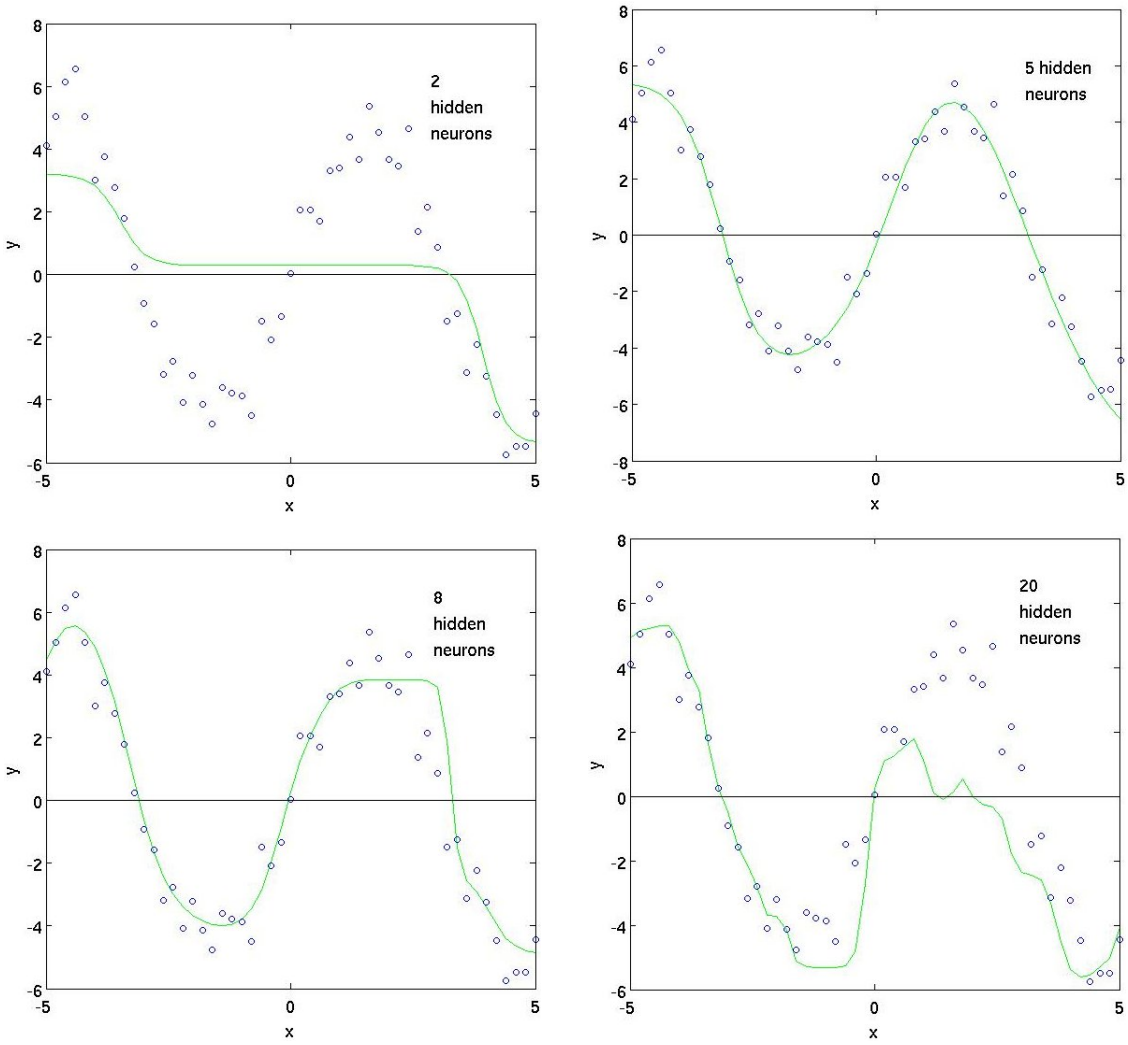


Figure 2.8: Fitting sine curve with different number of hidden neurons

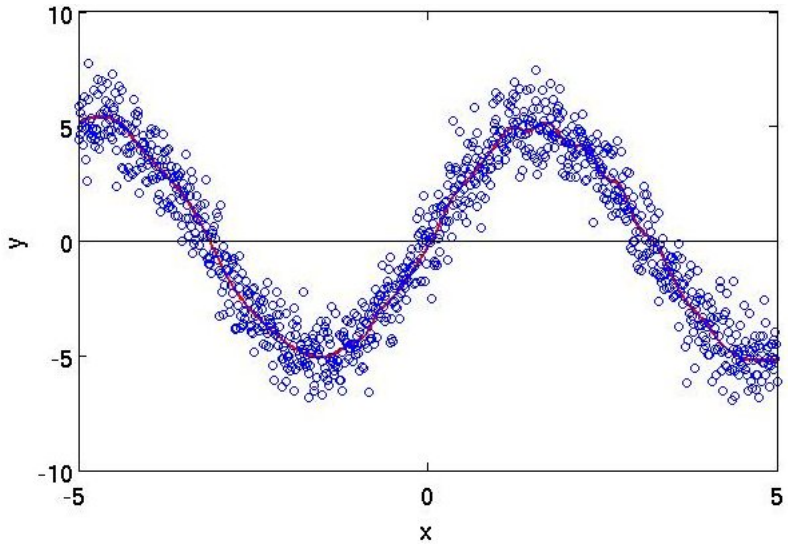


Figure 2.9: Fitting the same sine curve with 20 hidden neurons and 1001 datapoints

2.5 Applying Statistical Pattern Recognition to Heart Sound Analysis

Statistical pattern recognition is applied to heart sound analysis by considering cardiac cycles as objects to be recognized. As discussed in the previous chapter, certain types of cardiac disorder affect the operation of the heart in such ways that extra sound components (not one of the FHS) are generated in cardiac cycles. This means that cardiac disorder and extra heart sound components are directly linked and the detection of the sounds is equivalent to the detection of the diseases. Figure 2.10 shows a generic flowchart of heart sound analysis using ANN. It can be seen that the diagram has to branches. The left branch is for when the system is being trained, and the right branch is for when the system had already been trained and is testing a new unseen heart sound. The main difference between the two branches are that for the left branch, each of the operation (preprocessing, segmentation, feature extraction) are applied for each of the heart sound in the training set to get a set of feature vectors that will be used to train the ANN. On the other hand for the right branch, the operations are applied to only a single heart sound to get a single feature vector that become the input to a trained ANN and the output of the ANN indicates the type (normal or abnormal) of that heart sound.

2.5.1 Preprocessing

The first step of heart sound analysis is preprocessing. In the preprocessing stage, raw heart sound are decimated to a lower sampling frequency. Decimation means to low pass filter and then downsample. This is done because many sound recording program or digitizer use high sampling frequency to convert the analog sound signal into digital format. However, heart sound itself has relatively low frequency below 1000 Hz. Thus it is advantage

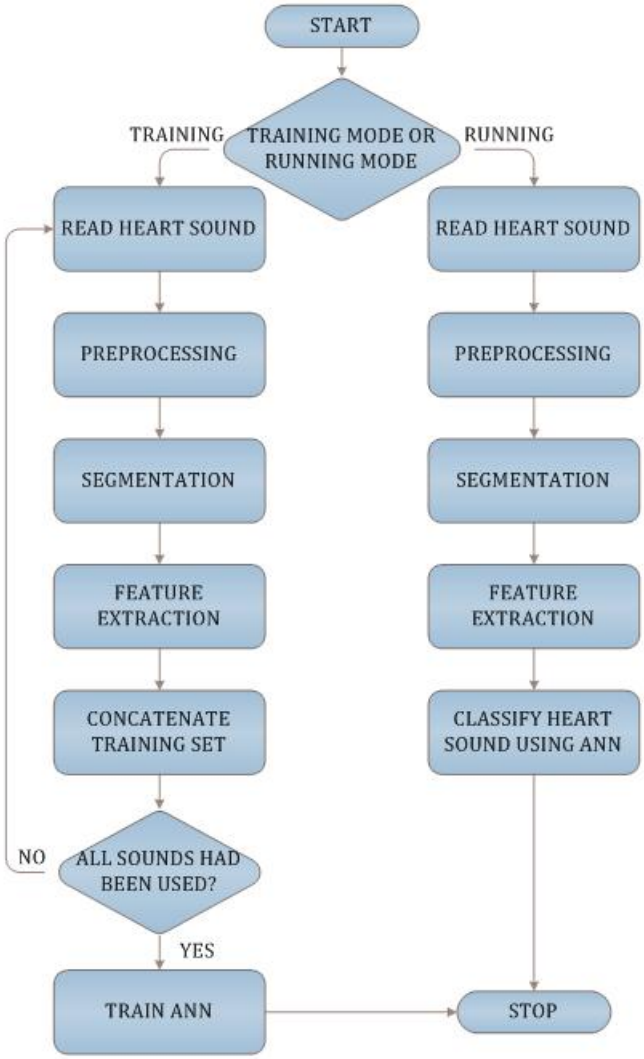


Figure 2.10: Heart sound analysis generic flowchart

to decimate raw heart sound in order to reduce the number samples that must be processed. Another operation that usually takes place during the preprocessing stage is noise removal. Heart sounds can be corrupted by two noise sources: ambient noise from the environment and noise from the stethoscope rubbing against the skin of the patient. The latter noise source is more like an artifact with very high intensity but short duration, therefore the portion of the signal that contain noise of this type can simply be discarded. For the ambient noise which is present throughout the entire signal, it must removed using some noise removal algorithm. In this study wavelet noise removal was used which is discussed in the next chapter. The pseudo code for preprocessing is given in algorithm 2.1. This work uses heart sounds with different sample frequencies, thus they all must be converted to a common sampling frequency before any processing can be done. This is accomplished in the preprocessing stage.

Algorithm 2.1 Generic preprocessing

1: **procedure** PREPROCESS(hs, Fs) ▷ hs is vector of heart sound and Fs is sampling frequency
2: x=convertFS(hs,4000) ▷ Convert f_s of hs to 4000 Hz and store it in x
3: x=filter(x,BPF) ▷ bandpass filter x using the filter coefficients stored in BPF
4: x=waveletNoiseRemoveal(x) ▷ Remove noise using wavelet
5: x=x-mean(x)/sd(x)▷ Normalized by subtract mean and divide by standard deviation
6: **end procedure**

2.5.2 Segmentation

The next step is segmentation, which means to determine the boundary of each cardiac cycle from a long heart sound signal. This step is necessary because cardiac cycles are the objects to be recognized. This is analogous to segmenting the face region from a picture of a person in facial recognition. Heart sound segmentation is based on the envelope signal and a generic algorithm is given in algorithm 2.2. The first step in segmentation is to calculate the envelope signal using any envelope detection method. Peaks of the envelope signal correspond to the FHS, which in the normal case can all be easily be detected by thresholding. One could then look at any group of three consecutive peaks and compare the distance between the first and second (p1-p2) against that of the second and the third (p2-p3). Using the assumption that systole is shorter than diastole, it is easy to identify the systole and thus S1 because it is always on the left side of systole. Having established the identify of just one peak, all other peaks can be identified by noting that S1 and S2 must alternate. Boundaries of cardiac cycles are then just the S1-S1 intervals.

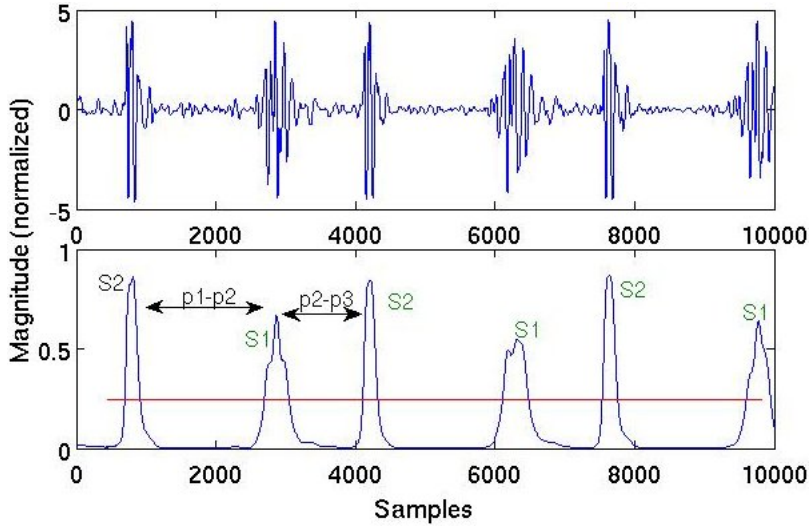
While segmentation of normal heart sound can be quite straight forward. The same cannot be said about abnormal ones. Figure 2.11 compares segmentation of normal heart sound and one with S3. Figure 2.11(a) show the segmentation of a normal heart sound. It can be seen that all the peaks can be picked up defining a fixed threshold shown by the red line. Also, every peak picked up by this process are those that corresponds to the FHS, no peak is missed and no extra peak is picked up. It is then a simple matter to pick up a series of three consecutive peaks and determine which one is S1 by comparing the p1-2 and p2-p3 intervals. From Figure 2.11(a), if one chooses the three left peaks to analyze, p1-p2 interval is greater than p2-p3 thus the second peak is S1. Once the identity of the second peak is known, all other peak can be identified by alternating S1 and S2 using the second peak as reference. It can be seen that in this case, all peaks are identified correctly as shown by the peak labels in Figure 2.11(a).

The difficulty in segmentation becomes apparent in Figure 2.11(b) which shows the case of abnormal heart sound with S3 (third heart sound that occurs just after S2). Again the three left peaks are analyzed they are actually S1, S2 and S3 respectively. The third peak

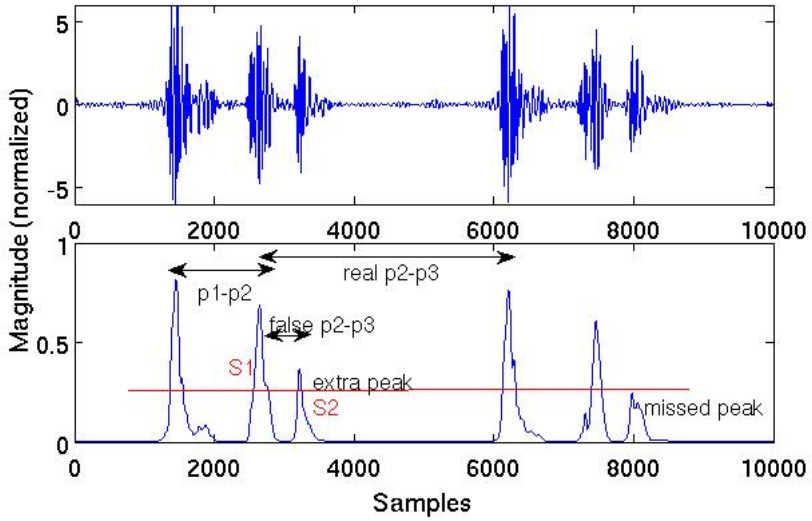
Algorithm 2.2 Segmentation by simple thresholding

1: **procedure** SEGMENTATION(x) ▷ vector x contains preprocessed hs
2: E = findEnv(x) ▷ find the envelope signal
3: T = some positive number
4: $peaks$ = blank array ▷ blank array to store position (x-axis value of each peak)
5: **for** $i \leftarrow 1, \text{length}(x) - 1$ **do** ▷ loop index runs from 1 length of x minus 1
6: **if** $x(i) \geq T$ and $x(i) > x(i+1)$ **then**
7: append i to $peaks$
8: **end if**
9: **end for**
10: $threePeaks$ = any 3 consecutive elements of $peaks$
11: $index$ = the element of $peaks$ that the first of the three peaks are taken
12: **if** $threePeaks(2) - threePeaks(1) < threePeaks(3) - threePeak(2)$ **then**
13: $peaks(index)$ = S1
14: **else**
15: $peaks(index)$ = S2
16: **end if**
17: $indexNeg = index$
18: $indexPos = index$
19: **while** $indexNeg - 1 \geq 1$ or $indexPos + 1 \leq$length(peaks) **do**
20: **if** $indexNeg \geq 1$ **then**
21: peaks(indexNeg-1) = opposite of peak(indexNeg) ▷ opposite means S1 becomes S2 and vise-versa
22: indexNeg = indexNeg-1
23: **end if**
24: **if** $indexPos \leq$length(peak) **then**
25: peaks(indexNeg+1) = opposite of peak(indexPos)
26: indexPos = indexPos + 1
27: **end if**
28: **end while**
29: **end procedure**

now is an extra peak S3 which does not corresponds to one of the FHS which includes only S1 and S2. The length of p2-p3 interval now changes, instead of being the distance between the second and the fourth peak like it should be without the extra third peak, it is now the distance between the second and third peak. Which is shorter than p1-p2 interval which leads to the false conclusion that the second peak is S1, and using this peak as reference leads to wrong segmentation result. The label for the first three peaks in Figure 2.11(b) are actually S1 S2 and S3, but instead it is determined by the segmentation process in algorithm 2.2 to be S2 S1 S2.



(a) All labels are correct because all the peaks are either S1 and S2



(b) Label of S1 and S2 are incorrect due to the extra third peak caused by third heart sound S3

Figure 2.11: (a) Segmentation of normal heart sound (b)That of heart sound with S3

It can be seen from that the difficulty is segmentation arises from extra components that may be present in cardiac cycles. These components generate extra peaks in the envelope signal that lead to the situation in Figure 2.11(b). At first it may seem that this can easily be solved by increasing the threshold so that no extra peaks are detected. But as the threshold is increased, there is a risk of missing some of the FHS as well. One could also set a low threshold to pick up every single peak and then try to determine which one are not FHS by considering the distance between each pair. If two peaks are closer together than a minimum distance (which is S1-S2 interval), then the smaller one is rejected. Typically S1 and S2 are about 50 millisecond apart but this vary from person to person. To calculate the minimum distant between each pair of FHS dynamically requires that the heart rate is known, but the heart rate cannot be calculated unless the heart sound had been segmented. Another

problem is that the relative size of S1-S2 (systole) and S2-S1 (diastole) intervals is not always guaranteed. Diastole shortens as the heart rate increases and eventually become that same length as that of systole. The heart rate at which this occurs varies from person to person. Many studies have proposed methods to solve heart sound segmentation problem. The list of which are given below and the details are discussed in the next chapter.

1. From the envelope signal, drop the smaller of the pair of peaks that are too close together (too close varies from 50 ms to 80 ms) or eliminating peaks that are too narrow (fixed value or a percentage of mean peak width) [7], [8]. The disadvantage of this approach is that it is sensitive to the envelope detection method. And if only a single extra peak remains after the peak-dropping process, it can cause incorrect segmentation result for every peak that come after it.
2. Assume that S2 has higher frequency than S1. Calculate two envelope signal using the Shannon energy of the wavelet approximation coefficient and details coefficient, respectively. The locations at which only the peaks of the first envelope occur are S1 and those at which peaks of both envelope occur are S2 [11]. This approach is based heavily on the assumption that S2 are of higher frequencies than S1. This difference is small and often not always true [2].
3. Eliminate components in a heart sound signal that are not FHS by thresholding of the discrete wavelet transform (DWT) coefficients. DWT coefficients that below the threshold value is considered noise and is reconstructed then subtracted with the original signal. The subtracted signal then becomes the input to the next iteration. The algorithm repeats a certain amount of time after which the output are just FHS, whose type are determined by the systole is shorter than diastole assumption [12]. This method is quite effective at eliminating extra components in heart sounds, but the difficulty in this method is setting the appropriate threshold for each iteration
4. Use Hidden Markov Model (HMM) in which each component of a cardiac cycle: S1, systole, S2, diastole is considered a state and the state sequence that most likely produced the observation (a vector of heart sound samples) is determined using HMM forward algorithm [13]. This method requires that all training heart sound be first segmented by hand in order to train the HMM. It does not require envelope detection or peak detection, at the expense of needing a human expert to manually segment training heart sound beforehand.

2.5.3 Feature Extraction

Feature extraction involves two parts; choosing an appropriate feature set and feature set optimization. The choice of a feature set is problem specific and is obtained mostly by experiment. Even in the area of heart sound analysis, there are so many different feature set that it is difficult to list them all. However, there is one feature set that get used repeatedly in many different studies which all reported good performance. This is the DWT coefficients. Due to the non-stationary nature of heart sounds, wavelet analysis which can be provide both

time and frequency information is particularly effective at extracting their important characteristics. The details of DWT is discussed in the next chapter. In this work, characteristics of envelope signal and DWT coefficients are used together to form the feature set.

Feature set optimization involve taking the feature set, eliminating some of the features and keeping only the subset that gives the best performance. This can be achieved by determining which features has more impact on the class label and ranking the features based on their class separability. This of course requires that the class label of the training set be known, which is true in this case. There are a number of procedures that can be used to rank features such as the f-test, t-test, and PCA discussed earlier in this chapter [5]. In this work, PCA is used because in addition to ranking the features, it also form new features as linear combination of the original ones. This is equivalent to projecting the datapoints formed by the feature vectors onto an axis that maximize their variance. Large variance of a feature variable give good class separability. This can be seen be considering Figure 2.2, in this case, the variance is low as can be seen by the two Gaussian curve close together. This means that the random variable x is confined within a relatively small range and hence its variance is relatively low. Given a particular value for the feature x, the probability $p(x|c_1)$ and $p(x|c_2)$ are comparable for most value of x. Compare this to Figure 2.1, in this case the two Gaussian curves are further apart so that the variance is larger. In this case for a given value of x it is more likely that $p(x|c_1)$ and $p(x|c_2)$ are significantly different in magnitude thus classification by selecting the maximum probability is more likely to be correct. As discussed in Section 2.2, the situation in Figure 2.1 is much easier to classify than that in Figure 2.2. This demonstrates the advantage of PCA over other feature ranking techniques, not only it reduce the number of features, it also project the remaining ones so that their variances are maximized, making it easier to classify based on them.

2.5.4 Training ANN

ANN training for heart sound analysis is the same as in any other statistical pattern recognition application. After each heart sound in the training set had passed through preprocessing, segmentation and feature extraction, one have training data which comprises of feature vectors with their respective class labels. Training starts by feeding each of the feature into the neural network and determine its output by propagating the input through the network as in Section 2.4.2 The output is subtracted by the class label to get the error value. Repeat this for all feature vectors in the training set to get the error due to each input. This is used to calculate the mean squared error (MSE) which is a measure of the deviation of the outputs of the output neurons to the desired ones. In ANN with multiple output neurons there are multiple MSE values, each for an output neuron. The MSE is a function of the connection weights of a neuron. The gradient of this function is calculated in order to find the direction along which the MSE decrease most rapidly and the connections weights are adjusted along this direction, multiplied by a scalar learning rate. This is called steep-

est descent backpropagation algorithm. A faster version of this algorithm called conjugate gradient algorithm is used in this research [14]. It is essentially backpropagation with speed improvement. It is designed to adjust the connection weights until the minima of the MSE is reached. However, there is no guarantee that this minima is a global minima. The algorithm may be stuck in a local minima in which case the performance of the ANN will be suboptimal. This is shown in Figure 2.12 showing some one dimensional imaginary error function for illustration. If the error function is initially at point A, moving in the direction of steepest decent will eventually arrive at a local minima. On the other hand, if the error function is initially at point B, in this case moving in the direction of steepest decent will arrive at the global minima. In actual case where the MSE is a function of many variables, t any be many local minima in which to be stuck at. Also, There is currently no network training algorithm that is guaranteed to find the global minima given any starting point. This means that final MSE after training, hence the performance a neural network is dependent upon the initial connection weights. There is no way one could know in advance the set of initial connections that will lead to a global minimum. In practice, the connection weights are initialized randomly, the network is trained and it performance noted. This is repeated many times and the network with the best performance is chosen for use.

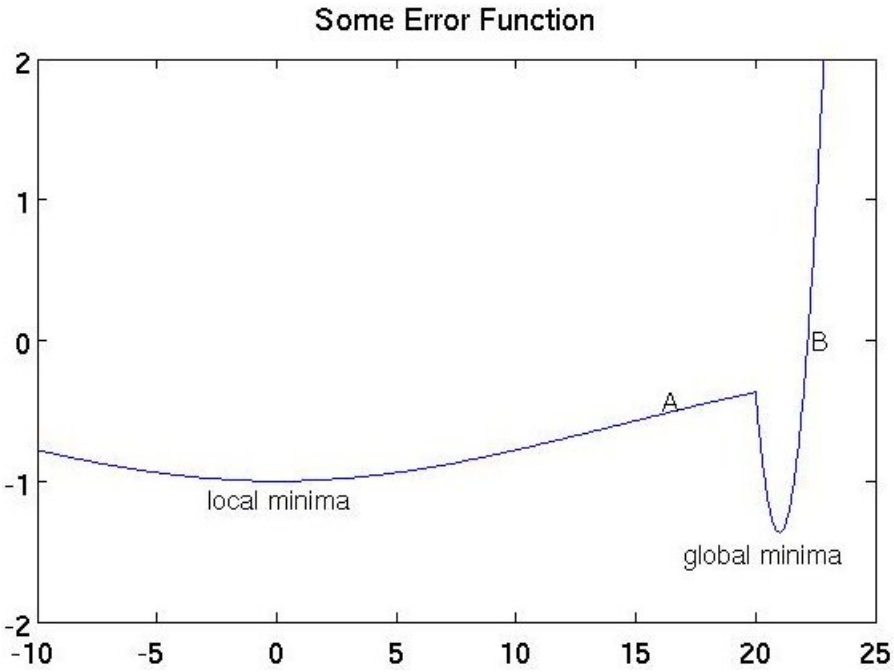


Figure 2.12: Comparison between local and global minima

2.6 Summary

This chapter had discussed statistical pattern recognition, upon which heart sound analysis is entirely based on. Heart sound analysis following this framework consists of preprocessing, segmentation, feature extraction and classifier training.

Preprocessing prepare raw heart sound signals for further processing, segmentation identifies the S1 and S2 sounds and determine the boundary of each cardiac cycle. Feature extraction encodes the characteristics of each cardiac cycle into feature vectors and the classifier is trained with a set of feature vectors representing the training set of heart sounds.

This chapter also discussed artificial neural network, which is one of the most popular classifier model in pattern recognition and in heart sound analysis in particular. Using ANN calls for many intricacy such as setting the number of hidden neurons, initializing and retraining ANN multiple times to make sure that it does not get caught in a local minima

The next chapter discusses the related literature each stage of heart sound analysis with a particular focus on segmentation because it is arguably the most challenging stage and it is this stage that this research focused on developing an alternative solution.

Chapter 3

Related Research

In this chapter, the literatures are organized according to the stages of heart sound analysis presented in the previous chapter. Segmentation is the most extensive part to highlight its difficulty because it is the most difficult stage and is the main problem addressed in this study.

3.1 Preprocessing

Preprocessing is mostly concerns with noise removal of the raw heart sound. This is because a typical physician's room can be quite noisy, especially in a small, busy hospital. Thus the recorded heart sound will contain a certain degree of noise, which makes noise removal an important part of heart sound processing.

3.1.1 Noise Removal of Heart Sounds by Using DWT coefficients

One popular noise removal technique is to use the discrete wavelet transform (DWT), a review of DWT is given in Section 3.3.1. It works by choosing a wavelet that is similar in shape to the desired clean signal so that the signal itself will be mapped to large wavelet coefficients and the noise mapped to small coefficients. By suitably thresholding the coefficients and reconstructing the signal, noise can be removed. There are however, many parameters in this process; the wavelet type, decomposition level, and the threshold levels. The study in [15] investigated noise removal of heart sound using MATLAB's wavelet toolbox. Test noisy heart sounds were processed using different parameters offered by the toolbox and the results compared by signal-to-noise ratio. They found that no single family of wavelet gives significantly better result than others. The optimum configuration for other parameters consists of 5 levels decomposition, soft thresholding, Steins unbiased risk estimate (SURE) threshold selection, and threshold rescaling using single estimation of level noise based on first level coefficients. In summary, noise removal is done using the command: wden(x,'SURE','mln','wname', 5), where x is the raw heart sound signal, and wname is the wavelet name. The result in [15] had been directly applied in this research.

3.2 Heart Sound Segmentation

This section discusses the literatures that focus on heart sound segmentation. This is arguably the most difficult stage in heart sound analysis, due to the complexity of heart sounds.

3.2.1 Segmentation Using Envelope Methods

Many of the proposed segmentation method is based on detecting the envelope of the heart sound or of one of its sub-band signals. One of the earlier study is [6]. Many later studies follow the same framework, with some modifications in the envelope detection methods such as using homomorphic filtering for envelope detection [16], or using a sub-band signal instead of raw heart sound [7], [8]. In [6], envelope of the heart sound is detected using the average normalized Shannon energy. Let x_{norm} denotes the normalized heart sound signal, then the average Shannon energy is calculated as

$$E_s = -\frac{1}{N}\sum_{i=1}^{N} x_{\text{norm}}^2(i) \times \log x_{\text{norm}}^2(i), \tag{3.1}$$

where N is the signal length in a 0.02 second period, its value depends on the sampling frequency used. Each sample of E_s is then normalized by subtracting with the mean of E_s and divided by the standard deviation. The resulting envelope waveform is then searched for the peaks by setting a threshold and picking every peak that is above the threshold value. An example of a heart sound and its envelope calculated using this method is shown in Figure 3.1.

The disadvantage of this method is that too many peaks are generated, thus ”peak conditioning” must be performed to account for them. First, extra peaks may have been picked up in the thresholding process, by ”extra” it means peaks that do not correspond to the FHS. These peaks may be caused by split heart sound, murmur, or noise. Identifying which peaks correspond to FHS and which do not is done by finding the intervals (distance) between each adjacent peaks and from the mean and standard deviation of the peak interval, the low-limit and high-limit of the interval size are set. Precisely how many standard deviation away from the mean are the low and high limit was not explained in [6], [7].

If an interval is shorter than the low-limit, then there is an extra peak that must be dropped. The criteria for deciding which to drop is based on the relative intensities of the two peaks that makes up the interval. The FHS are likely to be higher in intensity than other sounds, the when we have two peaks are close together than the low interval limit, the one with lower intensity should be dropped.

When an interval is larger than the high-limit, a lost peak is assumed be in between the two peaks forming this interval. The threshold is lowered incrementally until the lost peak

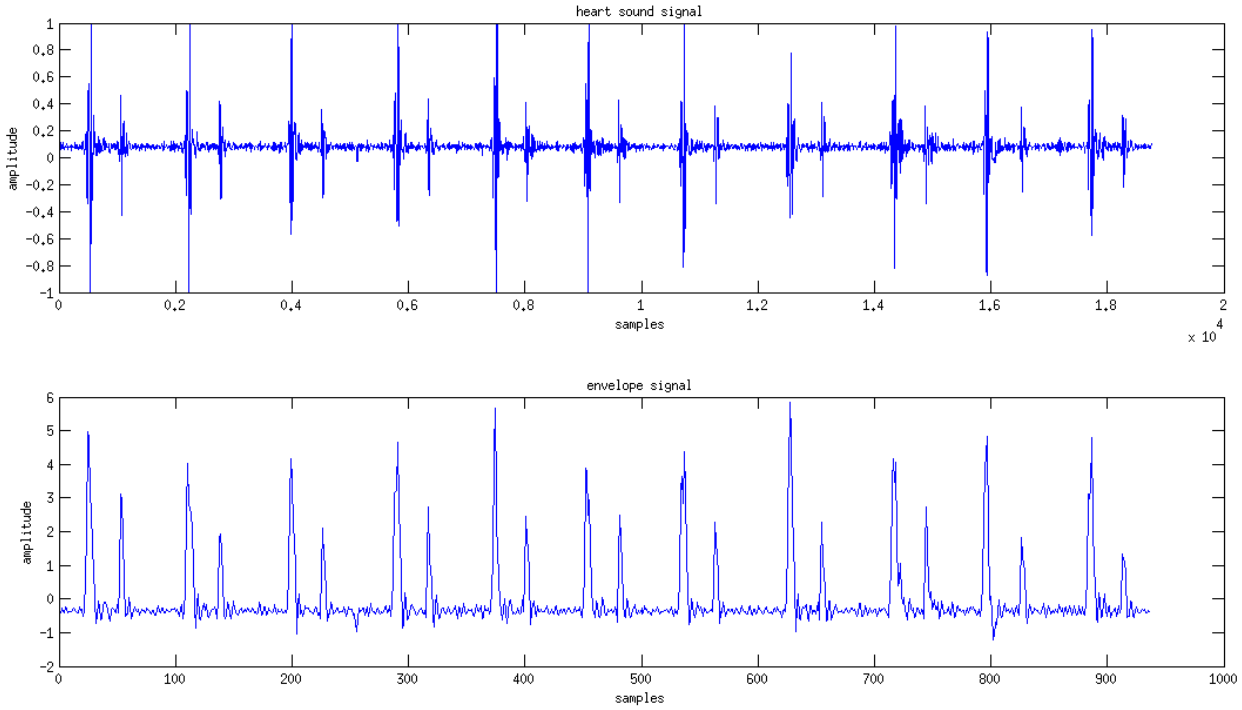


Figure 3.1: A clean heart sound and its Shannon energy envelope. The envelope signal is shorter than the original signal because it is calculated using sliding window in Equation 3.1

is detect. If more than one peak is detected, the criteria for rejecting extra peak is applied to them. When only the peaks belongs to the FHS are left, S1 and S2 are identified by noting that systole is shorter than diastole. The largest interval is identified, then intervals to its left and right are examined if they are consistent. That is, intervals must alternate between long and short one.

No procedure on how to calculate low and high limit of the peak intervals was explained. It could be assumed that they were probably calculated based on the mean and the standard deviation of the intervals themselves. This will only be valid if the number of extra peaks is not too many that they interfere with the statistics of the intervals. The appropriate value of the threshold was not given, and the rejection of extra peak and picking up of lost peaks involves many steps. Finally, identification of S1 and S2 is based on systole being shorter than diastole, which is not considered to be always true in the medical literature[4]. Overall, [6] provided a framework for heart sound segmentation for many later studies. The main disadvantage of their method is the smoothness of envelope detection using Shannon energy method. The envelopes determined using this method for heart sound with extra component, especially those with murmur sounds are very erratic that they do not look like envelope signal at all but rather like the original heart sounds. In other words, segmentation using this method fails for heart sound with murmur.

Some later studies such as [16] uses an improved envelope detection technique based on *homomorphic filtering*. Let $x(n)$ denotes the heart sound signal, then its energy can be

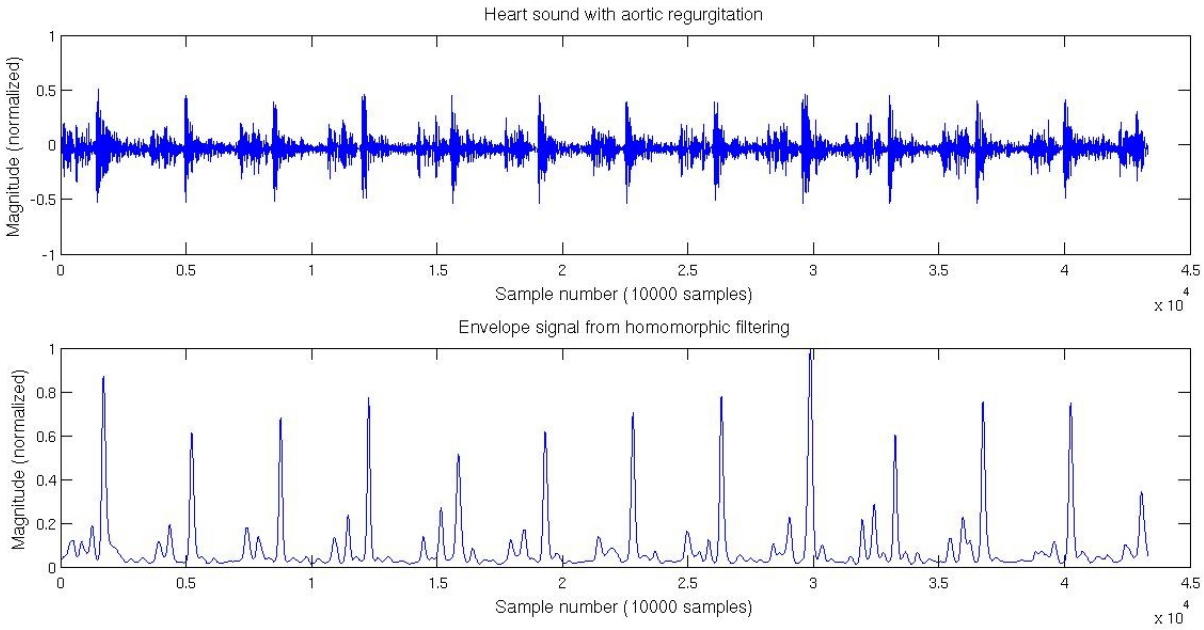


Figure 3.2: Heart sound with aortic regurgitation and its envelope

expressed as

$$e(n) = x^2(n) = a(n)f(n), \tag{3.2}$$

where $a(n)$ is the slow varying part and $f(n)$ is the fast varying part, similar to amplitude modulated (AM) signal. Multiplication can be converted to addition by taking the logarithm, then

$$z(n) = \log(e(n)) = \log(a(n)) + \log(f(n)), \tag{3.3}$$

which gives the combination of the slow and fast varying components additively. The fast varying component $f(n)$ can then be removed by low pass filtering,

$$z_1(n) = L\left[z(n)\right] = L\left[\log(a(n))\right] + L\left[\log(f(n))\right] = \log(a(n)) \tag{3.4}$$

and the final step is to remove the logarithm by taking exponent,

$$\exp(z_1(n)) = \exp(\log(a(n))) = a(n) \tag{3.5}$$

A heart sound with aortic regurgitation (which is a type of heart murmur associated with leaking of the aortic valve) and its envelope signal calculated using homomorphic filtering method is shown in Figure 3.2.

The envelope quality is much better than using Shannon energy, there are high peaks that corresponds to the S2 sounds. However, S2 sounds are diminished in the envelope signal due to the effect of the murmur. Also, the envelope is so smooth that peaks that corresponds to murmur can disappear altogether, as shown Figure 3.3 for a heart sound with severe aortic stenosis. Figure 3.3 on the top panel clearly shows this is a strongly

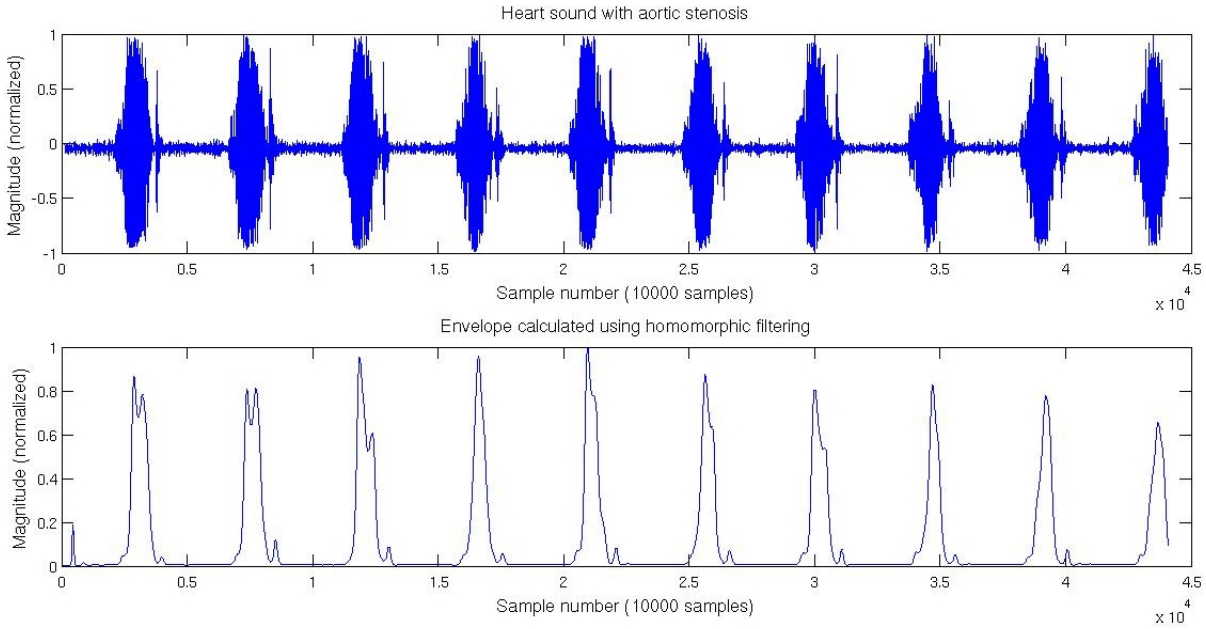


Figure 3.3: Heart sound with severe aortic stenosis and its homomorphic envelope calculated from Equation 3.2 through 3.5

abnormal heart sound, however, the bottom panel looks like the envelope of normal heart sound. Clearly any segmentation based on the envelop in Figure 3.3 will yield false cardiac cycles boundaries.

3.2.2 Segmentation Using Time-Scale Representation

The study in [17] used a type of time-scale representation called wavelet to obtain the envelope of heart sound. Wavelet is a transform technique to converts a signal from time domain to scale-frequency domain. A general formula for wavelet transform for continuous signals $f(x)$ is given by [18]:

$$W(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} f(x)\psi^* \left(\frac{x-b}{a} \right) dx, \tag{3.6}$$

where W is the wavelet transform domain representation, the parameters a and b are called the *scale* and *shift*, respectively. The function ψ is called the *mother wavelet*. A mother wavelet is a function with small duration around the origin that resembles a small wave as shown in Figure 3.4. In computing the wavelet transform, the mother wavelet ψ is scaled and shifted according to the parameters a and b, multiplied with the signal $f(x)$ and integrated throughout the duration of the signal. The scale a controls how compressed or expanded the mother wavelet becomes, high value of a means that the mother wavelet is expanded i.e., low frequency. On the other hand, low value of a means that the mother wavelet is compressed, i.e., high frequency. Thus the scale parameter is said to encodes the frequency information

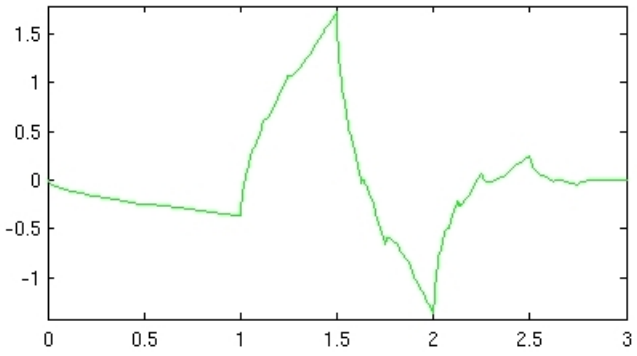


Figure 3.4: A mother wavelet ψ

of the signal. The shift parameter b determines the location of the mother wavelet with respects the signal, thus it encodes the time information of the signal. A 2D color plot of the magnitude of W for all values of a and b thus resembles the spectrogram of the short-time Fourier transform and is aptly called the *scalogram*. Envelope detection based on wavelet is based on the similarity between a type of mother wavelet called the Morlet wavelet and The FHS as shown in Figure 3.5. Let $X(k)$ be a vector of length N that represents the heart sound signal, then one can calculate a M by N matrix of wavelet coefficients Y where M is the number of scale bins using the following Equation

$$Y(m,n) = \sum_{k=1}^{N} \psi_m(k-n)X(k), \tag{3.7}$$

which is essentially a discrete version of Equation 3.6 with the normalizing factor omitted. The scale parameter had already been absorbed into ψ_m, given by,

$$\psi_m(k) = \pi^{-1/4} \exp(\frac{-k^2}{2a_m^2}) \exp(j\omega_m k), \tag{3.8}$$

where $\omega_m = \frac{\omega_0}{a_m}$ is the *center frequency* associated with ψ_m and $\omega_0 = 5$ rad/s is the center frequency of the Morlet mother wavelet. The center frequency is the frequency at which the magnitude of the Fourier transform of the wavelet is maximum.

In [17]The scale parameter a_m was chosen such that the resulting set of ψ_m logarithmically covers the frequencies that corresponds to the typical frequency range of FHS. Applying a specific set of scales a_m together with Equation 3.7 gives a 2D representation of the heart sound Y. Then the envelope signal E was calculated from Y using

$$E(n) = \sum_{m=1}^{M} |Y(m,n)|^2, \quad m = 1, 2, ..., N. \tag{3.9}$$

Due to the similarity between ψ_m and the FHS, large coefficients of Y corresponds to the time location where a FHS is present, which is the same locations that the high peaks on

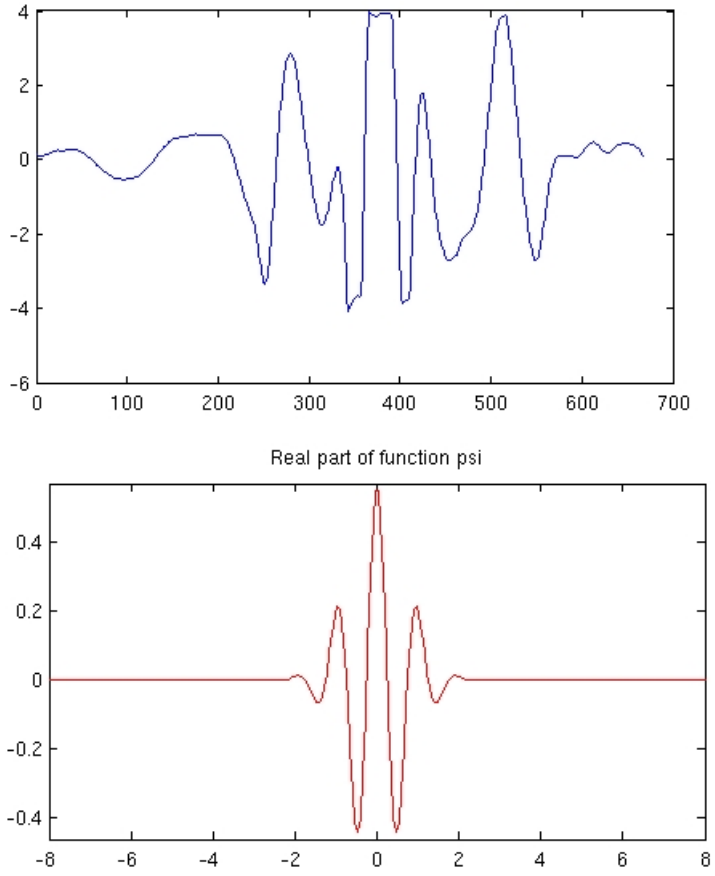


Figure 3.5: A FHS and Morlet wavelet

$E(n)$ will occur, even if it is buried in noise. This procedure is equivalent to calculating the scalogram of a heart sound and taking the sum of each column. The envelope of the same heart sound with aortic regurgitation calculated using Morlet wave is shown in Figure 3.6. There are high peaks that corresponds to the S2 sounds, however, even the wavelet method was unable to clearly locate the S1 sounds which had been contaminated by the murmur. The envelope signal around S1 appears to be squashed down and spread out into many small peaks and the location of S1 is uncertain.

3.2.3 Conclusion on Segmentation

One of the main difficulty in heart sound analysis is in the segmentation stage. Segmentation is based on envelope analysis. The envelope signal of a heart sound is calculated, the peaks of this envelope should corresponds to S1 and S2 (together referred to as fundamental heart sound, FHS). For healthy heart sound this is the case, however in abnormal sounds that has extra components in their cardiac cycles, it is easy to confuse the peaks that corresponds to the extra components as the FHS. There are methods in the literature that deal with this problem by dropping peaks that are too close together and lower the peak detection threshold

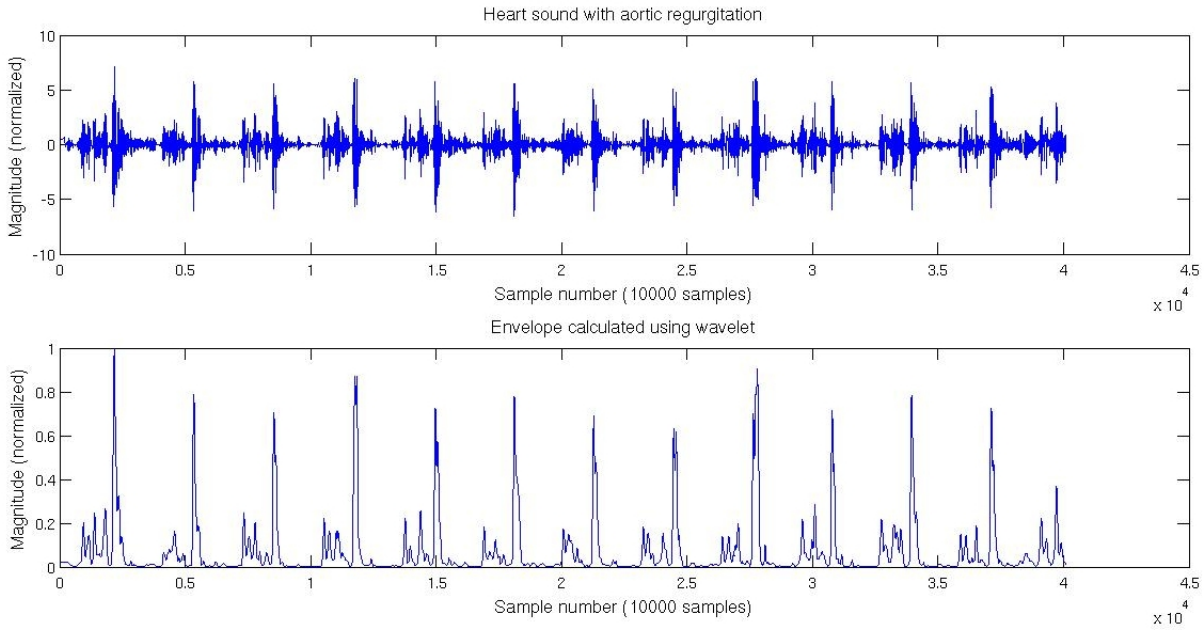


Figure 3.6: A cardiac cycle and its envelope calculated using wavelet coefficient

when peaks are too far apart. However, ”too close” and ”too far” are based on the statistics of the peak intervals, which assumes that most peaks are detected correctly. There is no way to determine without help of an external reference that all cardiac cycles are detected correctly and using miss-detected cardiac cycle in feature extraction leads to erroneous feature vectors and eventually lowers the classifier's performance.

This section reviews the literature on heart sound segmentation. Segmentation is based on analyzing the envelope of the heart sound signal, studies that used Shannon energy to calculate the envelope are[6], [7]. This method yields envelope signals that contain excessive number of peaks for heart sound with murmur. A newer method is [16], where the envelope signal was calculated using homomorphic filtering. This method gives a very smooth envelope but can lead to false segmentation on some type of heart sounds. A method based on wavelet is [17], which takes advantage of the Morlet wavelet's resemblance to the FHS to calculate the envelope signal. This method was able to obtain high quality envelope signal, however the locations of some FHS were difficult to determine accurately due to murmur causing the envelope to spread out into many small peaks.

In all segmentation methods, the type of each FHS is determined by using the assumption that systole is shorter than the diastole. Thus, given three consecutive FHS peak, their types can be identified by comparing the interval between the first peak and the second peak with the interval between the second and third peak. If the first interval is larger, then the first FHS is S2, otherwise, it is S1. The difference between the lengths of systole and diastole however, decrease with increasing heart rate. Coupled with deviations in determining the location of each FHS, this can lead to incorrect segmentation results, which will affect the

feature extraction process. This is the reason why in this research, an alternative to performing segmentation is proposed.

3.3 Feature Selection

The choice of features reported in the literature are quite varied, thought they roughly fall into two groups. The first is to use the time and frequency information (not joint time-frequency distributions) extracted directly from the heart sound signal. The second is to use wavelet coefficients as features.

An example of a feature set that falls the first group is [7], in which the features are:

1. The peak intensity and peak timing of S1 compared to the start of S1 and the duration of S1 itself.
2. The duration of S2.
3. Peak intensities of the aortic component of S2 (A2) and that of the pulmonic component (P2). The splitting interval, and the onset of A2 compared to the beginning of S2.
4. The duration and the three largest frequency components of the systole and the shape of the systolic murmur.
5. The duration and the three largest frequency components of the diastole and the shape of the diastolic murmur.

The rational behind the first two features is that murmurs tend to overlap with the FHS, altering their durations. The third feature is based on the fact that split S2 is an indication of aortic stenosis and atrial/ventricular septal defect. The last two features are decided to detect the murmurs themselves, since murmurs generally increase the frequency and alter the shape of the systolic/diastolic signal.

Clearly, such a set of features is empirical in nature. In general, one would chose features that are good indicators of the disorders to be detected. However, some features are more difficult to extract than others, such as the splitting interval of S2. The extraction of this feature relies on accurate envelope detection, as an envelope that is too smooth will make split S2 appears to be a single peak instead of two. Thus a feature should not only be good indicator of pathology, but it has to also be easy to extract. It can be seen that choosing an appropriate feature set this way is a trial and error process, but the resulting feature set was smaller compared to another approach to feature selection, which is discussed in the next section.

3.3.1 Discrete Wavelet Transform

This is the second approach to feature selection. There are many studies that used features obtained from the DWT coefficients or use the coefficients themselves as features.

DWT is a very powerful signal representation technique that allows for simultaneous time-frequency information and finds extensive use in heart sound analysis and it signal processing in general. DWT is a time-frequency representation of signals [18]. It provides similar information as the scalogram in Equation 3.7 but it is designed specifically for discrete-time signal.

A signal x can be decomposed into it DWT coefficient according to the following tree in Figure 3.7. This tree is also known as *filter bank*

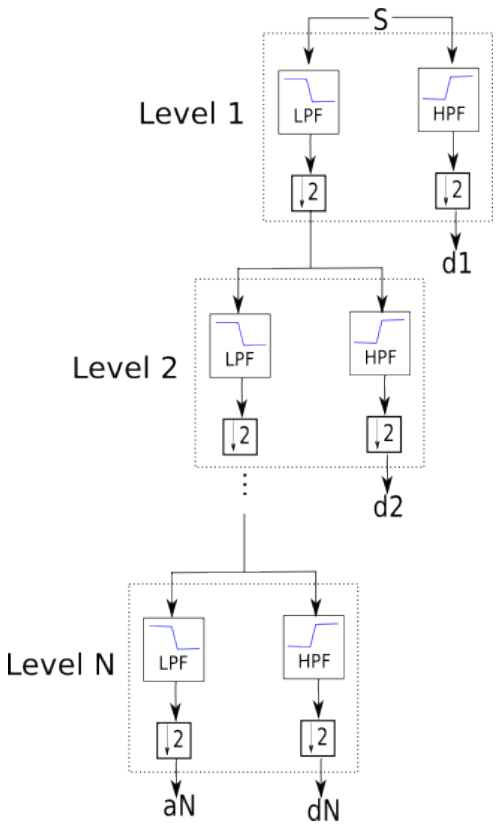


Figure 3.7: A DWT decomposition tree

The signal S is passed through *half-band* low pass filter and high pass filter. The output of these filter has only half the frequency information as the signal S but the same number of samples, to maintain the same frequency resolution, the outputs of the filters must be down-sampled by a factor of two, an indicated in Figure 3.7. The down-sampled output of the high pass filter is called the *detail coefficients*, abbreviated as d, and the number after d indicates the level of signal decomposition. The down-sampled output of the low pass filter can be passed through more pairs of half-band filters and the signal is further decomposed, each level of decomposition is indicated by the dotted box in Figure 3.7. At each level, the bandwidth of the signal is halved and the lower half is decomposed further, the frequency

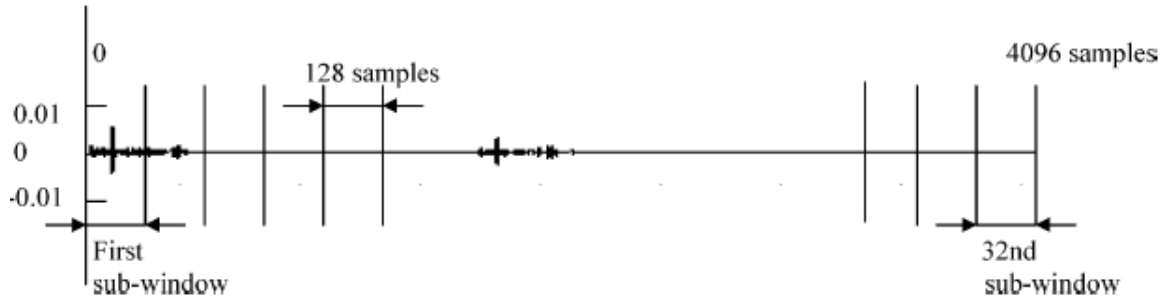


Figure 3.8: Feature extraction from d2 wavelet coefficients

resolution thus increases as the signal traverses down the tree, since each level represents half the frequency band as the previous one. On the other hand, the time resolution is decreased as the signal traverse down the tree since less coefficients are used to describe the signal. Thus DWT is a variable-scale representation of signal, lower frequencies have high frequency resolution but low time resolution, while higher frequencies have low frequency resolution but high time resolution.

3.3.2 DWT Coefficients as Features

In [19], the segmented heart sound is decomposed in seven levels using fourth order Coifman wavelet. Then the d1 through d4 coefficients are discarded. The remaining coefficients are concatenated together and normalized. This forms a feature vector that is 256 elements long. This size of feature vectors is quite large and obviously had some redundancy which should have been removed. Using features vectors which are 256 elements long means that the size of the ANN classifier is consequently large, and a lot of data was needed to train such a large network. In [8], heart sounds were sampled at 5512.5 Hz and a cardiac cycle is always taken to be 4096 samples long. Each cardiac cycle is decomposed to six levels using second order Daubechies wavelet. The d2 coefficients, which represented the frequency band of 1378-2756 Hz, were divided into 32 equal-length and non-overlapping portions. The signal power of each portion was taken as a feature, resulting in a feature vector that is 32 elements long. This is illustrated in Figure 3.8. The advantage of using DWT coefficients as features this way is that no prior knowledge of how each cardiac disorder affects the heart sound is needed and the length of feature vectors is not too large, even though some redundancy was still present.

Some other feature sets are; combination of time, frequency, and time-frequency distributions [20]. Mel-frequency cepstral coefficient [21]. However, the DWT features in [8] is the most common across many studies.

3.4 Classifier Optimization

Some studies focused mainly on the improvement of the basic ANN classifier. The studies in [7], [19] used basic feed-forward neural network such as that in Figure 2.4, trained by back propagation algorithm in which the network structure needed to be chosen prior to training the network. The study in [16] used growth and learn (GAL) network and [8] compared the performance of GAL and linear vector quantization (LVQ) network [10]. The objective of these special neural network training scheme is to optimize the network structure by minimizing the number of neurons. More recently, [9] proposed a method to systematically optimize the ANN structure, reducing the size and complexity of the network. The feature used in the study was the same as that in [8]. The performance of a non-optimized feedforward ANN (32 inputs, 115 hidden and 13 output neurons) was 95.192 precent, while that of the optimized network (26 input, 25 hidden and 8 output neurons) increased to 99.279 percent. The study did not however explain how the performance was calculated and a performance improvement of 4 percent may be the result of reducing the network size, resulting in better generalization due to less over-fitting. Regardless, the result of [9] strongly demonstrated that using the signal power of d2 wavelet coefficients is a good feature choice, and the basic feed-forward ANN is already a very good classifier. Thus in this work the regular feedforward ANN with experimentally chosen number of hidden layer neurons are used.

This chapter had reviewed the related literature on heart sound analysis. Optimum noise removal was the first topic reviewed, followed by segmentation of heart sound using different envelop detection methods. In feature selection, two main approaches to building a feature set reviewed and compared. Finally, the chapter ends with the discussion of improving performance the ANN classifier.

Chapter 4

Proposed Heart Sound Analysis Method

This Chapter presents the proposed heart sound analysis method for this thesis. It is organized according to the flow of heart sound analysis and presents the details of each stage.

4.1 The Training Set

The training set was the starting point in algorithm development, in this study, a total of 57 heart sounds comprises the training set. They were recorded using a regular electronic stethoscope as well as obtained from various Internet sources[22], [23], [24], [25]. These heart sounds were recorded using different equipments under different conditions, thus they are more challenging than homogeneous set in which all heart sounds were recorded with the same equipment in the same condition. The number of each type of heart sound is given in Table 4.1, where the ”(m)” after the heart sound’s name indicates that it is a heart sound with murmur. The training set is arranged into two groups: ”normal” and ”abnormal” with corresponding class labels ”0” and ”1” respectively.

These heart sounds were from different sources, which means that they were collected from different patients, by different doctors using different equipments. Also, it is common practice of doctors to change the auscultation site slightly, according to what kind of sound is being heard, because different types of abnormal heart sound can be best heard from different places on the patient’s chest. Therefore, this training set accounted for the variability in patients, stethoscope type, and auscultation site. A classifier that can correctly classify this set could be considered to be robust.

4.2 Overall Flow

The overall flow for heart sound analysis procedure used in this thesis is shown in Figure 4.1. PCA means principal component analysis, which was introduced in Section 2.1.2 and the details is given in Section 4.6 The preprocessing stage prepares each raw heart sound signal for further analysis. Envelope detection computes the envelope signal of each heart sound using wavelet coefficients. The length of cardiac cycles were calculated using the

Table 4.1: Heart sounds used in the training set

Heart Sound Type	**Number of Files**
Normal	12
Third Heart Sound	4
Fourth Heart Sound	3
Ejection Sound	2
Systolic Click	2
Summation Gallop	1
Opening Snap	2
Split S2	4
Aortic Regurgitation (m)	6
Aortic Stenosis (m)	6
Mitral Regurgitation (m)	6
Mitral Stenosis (m)	5
Pulmonary Stenosis (m)	4

autocorrelation of the envelope signals. Segments of both of the heart sounds and envelope signals with lengths equal to five cardiac cycles were segmented. Three features; number of peaks, average peak interval and sum of all samples, were extracted from the envelope segments and 32 additional features formed by energy of DWT coefficients were extracted from the heart sound segments. These features were concatenated together to form feature vectors with 35 elements. The individual feature vector is denoted as $\mathbf{x_i}$, each with its corresponding class label d_i. The training set is denoted as $\mathbf{X} = \{\mathbf{x_i}, \mathbf{d_i}\}_{\mathbf{i=1}}^{\mathbf{Q}}$. PCA analysis was performed on $\mathbf{X}$ to reduce the dimension of the feature vectors by forming linear combinations of each feature variable. Let $\mathbf{X}$ now denote the training set that had passed through PCA, $\mathbf{X}$ was sampled randomly Q times to obtain a new training set and this in turn was repeated M time to generate M new training set $\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_M}$, each with the same number of feature vectors as the original feature set $\mathbf{X}$. This process is known as *bootstrapping*. Using these new training set, M neural networks are trained, each with its own training set $\mathbf{X_i}$, then neural networks form a *bagging* classifier [26]. Given an unknown feature vector, the bagging classifiers make a decision on its class label by *voting* on their individual outputs.

4.3 Preprocessing

The preprocessing involves low pass filtering with an FIR low pass filter with cutoff frequency at 600 Hz. Then re-sampling all heart sounds to a common sampling frequency at 4000 Hz. Next was noise removal using DWT method as in [15]. The final preprocessing step was to normalize each heart sound by subtracting by its mean and dividing by its standard deviation. The pseudo code for preprocessing was given in Algorithm 2.1, it is repeated

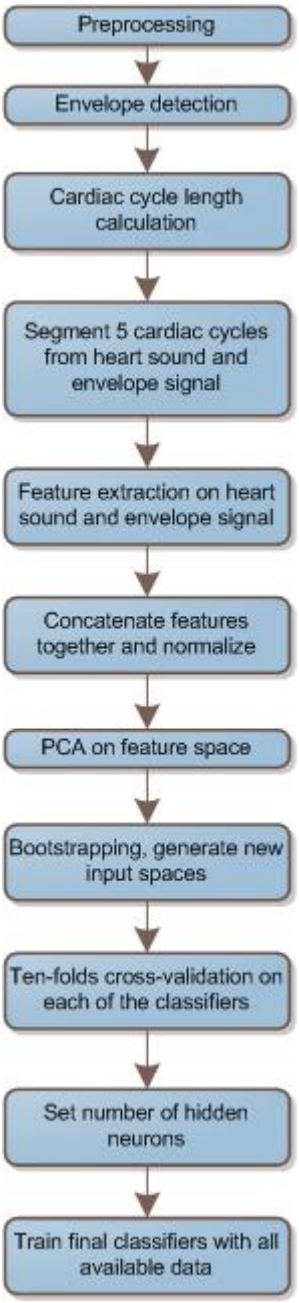


Figure 4.1: Heart sound analysis flow

again in Algorithm 4.1 for easy reference.

4.4 An Alternative Segmentation

The major contribution of this research is the development of an alternative scheme to replace the segmentation step. It is no longer necessary to perform segmentation of heart sound in the traditional sense. That is, it is no longer necessary to detect and determine the type of each FHS then form cardiac cycles using S1-S1 intervals. Under this scheme there

Algorithm 4.1 Preprocessing

1: **procedure** PREPROCESS(hs, Fs) ▷ hs is vector of heart sound and Fs is sampling frequency
2: x=convertFS(hs,4000) ▷ Convert f_s of hs to 4000 Hz and store it in x
3: x=filter(x,LPF) ▷ bandpass filter x using the filter coefficients stored in vector LPF
4: x=waveletNoiseRemoveal(x) ▷ Remove noise using wavelet
5: x=x-mean(x)/sd(x)▷ Normalized by subtract mean and divide by standard deviation
6: **end procedure**

are just three main steps that must be performed for each preprocessed heart sound:

1. Envelope detection
2. Cardiac cycle length calculation
3. Extract segments with lengths equal to five cardiac cycles from heart sound signal and its envelope

4.4.1 Envelope Detection

The envelope detection method in [17] was used in this research. That is, given the preprocessed heart sound $x(k)$ the envelope signal E is calculated by,

$$\psi_m(k) = \pi^{-1/4} \exp(\frac{-k^2}{2a_m^2}) \exp(j\omega_m k) \tag{4.1}$$

$$Y(m,n) = \sum_{k=1}^{N} \psi_m(k-n)x(k) \tag{4.2}$$

$$E(n) = \sum_{m=1}^{M} |Y(m,n)|^2, \quad m = 1, 2, ..., N., \tag{4.3}$$

where the scale parameters a_m were determined by first enumerating all the frequencies from 10 to 300 Hz, with 8 bin per octave. That is, the frequency band between 10 to 20 Hz was divided into 8 bins, and from 20 to 40 Hz into another 8 bins, then from 40 to 80 Hz into yet another 8 bins. This continued until 300 Hz and the frequency bins are now located logarithmically. The corresponding scale of each frequency bin was determined by plotting the function MATLAB function scal2frq as shown in Figure 4.2 and tracing the plot to find the appropriate scale values ψ_m. The function scal2frq links scale(s) to frequency using,

$$s = \frac{F_c}{F\Delta}, \tag{4.4}$$

where F is the frequency, F_c is the center frequency of the mother wavelet which is 5 rad/s for Morlet wavelet, and Δ is the sampling period. The frequencies and their corresponding scales are summarized in Table 4.2

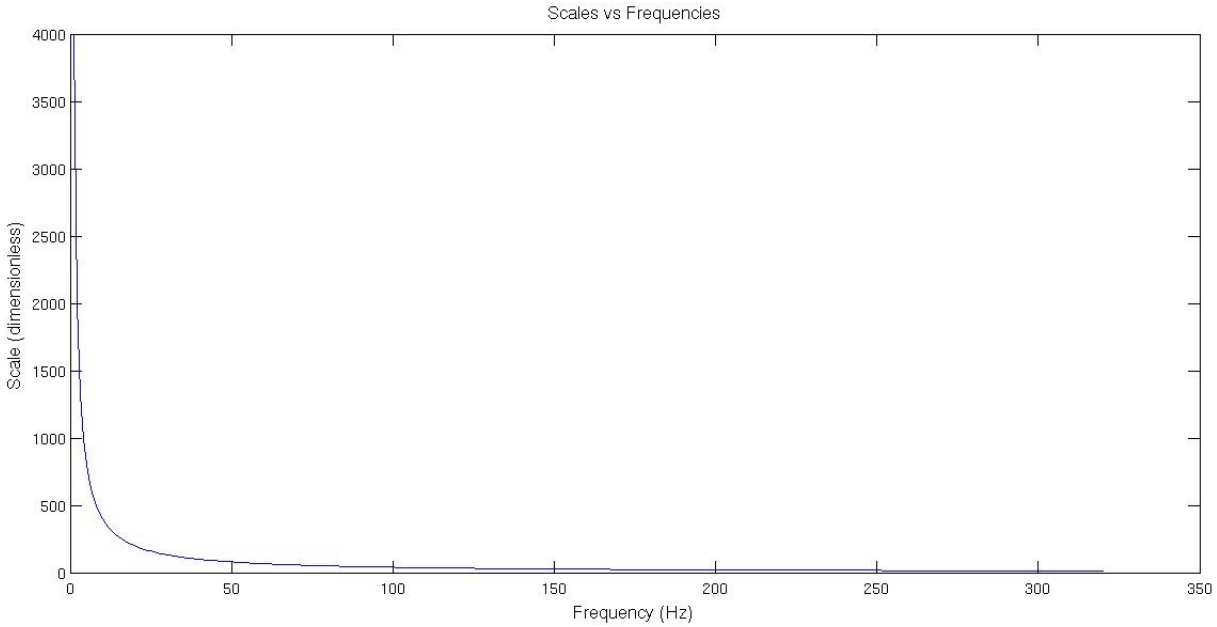


Figure 4.2: Plot of scales vs frequencies for Morlet wavelet

Table 4.2: Scales and frequencies used in wavelet-based envelope detection

Freq	**Scale**	**Freq**	**Scale**	**Freq**	**Scale**	**Freq**	**Scale**
10	661	25	265	60	110	140	47
11.25	588	27.5	241	65	101	150	44
12.5	529	30	220	70	94	160	41
13.75	481	32.5	204	75	88	180	37
15	441	35	189	80	82	200	33
16.25	407	37.5	176	90	73	220	30
17.5	378	40	165	100	66	240	28
18.75	353	45	147	110	60	260	25
20	330	50	132	120	55	280	24
22.5	294	55	120	130	51	300	22

There scales in Table 4.2 together with Equation 4.1 throught 4.3 were used to calculate the envelope signal of each heart sound. Each envelope signal was normalized so that its maximum magnitude is unity.

4.4.2 Cardiac Cycle Length Calculation

The length of the cardiac cycle of a heart sound needs to be determine because it can be used as a ”measuring stick” such that the same number of cardiac cycle and hence the same amount of information can be segmented from heart sounds with different heart rates.

Determining the length of a cardiac cycle is based on the quasi-periodicity of heart sounds. Strictly speaking, heart sounds are not periodic since no two cardiac cycles are exactly the same, however, they have high degree of similarity such that there will be peaks in the autocorrelation function of the envelope signal due to cardiac cycles lining up when the signal was shifted to calculate the autocorrelation. The distance from the beginning of the autocorrelation signal to the first peak corresponds to the length of cardiac cycle for that heart sound. The autocorrelation function is given in Equation 4.5, where N is the length of the signal whose autocorrelation is to be calculated. Autocorrelation sequence of a real sequence x with length N is a real even-symmetric sequence with length $2N-1$. Since R_{xx} is even-symmetric, only the right hand side ($m \geq 0$) need to be considered.

$$R_{xx}(m) = \begin{cases} \sum_{n=0}^{N-m-1} x(n+m)x(n) & m \geq 0 \\ R_{xx}(-m) & m < 0 \end{cases} \tag{4.5}$$

The envelope signal was used to calculate the autocorrelation rather than the raw heart sound because the peaks of the resulting autocorrelation function was much more clear and also because one is only interested in the period of the cardiac cycle which can be more easily inferred from the envelope signal.

The autocorrelation function was searched for the location that the maximum occurs, limited within the window of 1000 samples to 5000 samples from the start of the signal so that the global peak at $m = 0$ is not considered. At the sampling frequency of 4000 Hz, this window covers the range of cardiac cycle period of 0.25 to 1.25 second, or equivalently 48 to 240 beat per minutes which covers all possible human heart rates. An envelope signal and its right-sided autocorrelation function is shown in Figure 4.3

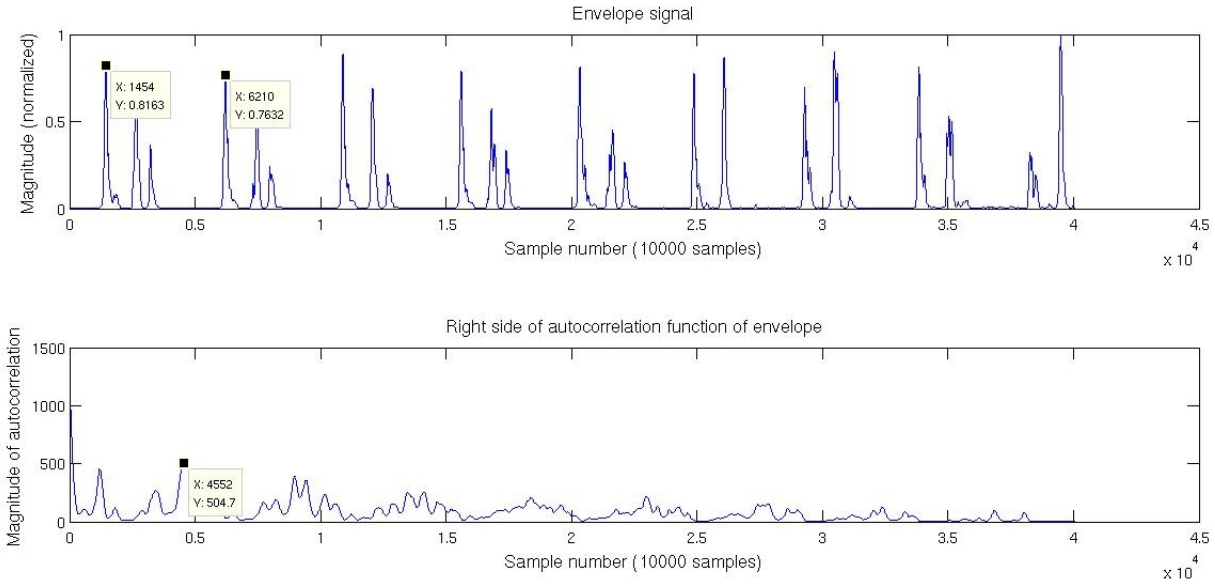


Figure 4.3: An envelope signal and it right-sided autocorrelation function

It can be seen by visual inspection of Figure 4.3 in the top panel that the length of the cardiac cycle is $6210 - 1454 = 4756$ samples, which is roughly the same as the cardiac cycle length determined by searching for the maximum within the window of 1000-5000 samples from the beginning of the autocorrelation signal shown in the bottom panel of Figure 4.3 which is 4552. It can be seen that the calculated cardiac cycle length is off by about 200 samples from the actual value, which at $f_s = 4000$ Hz equal to just 0.05 s. Once the length of cardiac cycle is known, in the next step (step 3 of the proposed method) a segment which starts from the beginning of the signal and ends at the length of five cardiac cycles was "cropped out" from the heart sounds and their envelope signals. Cropping five cardiac cycles instead of just one averages out the possibility of classification error due to a cardiac cycle being significantly different from other cycles. The choice of five cycles is a trade-off between this averaging effect and the length of signals needed to perform the analysis. The pseudo code for the proposed segmentation method is given in Algorithm 4.2

Algorithm 4.2 Proposed segmentation method

1: **procedure** SEGMENTATION(x) ▷ x is a preprocessed heart sound
2: s=scales in table 4.2 ▷ s is a vector of scales to calculate the wavelet coefficients
3: $E = \mathrm{env}(x, s)$ ▷ Find the envelope according to Equation 4.1 through 4.3
4: $r = \mathrm{autocorrelation}(E)$ ▷ find the autocorrelation of the envelope using Equation 4.5 for $m = 0, 1, 2, ..., N$ where N is the length of x
5: $L = max(r(1000 : 5000))$ ▷ find the maximum of r in the range $r(1000)$ to $r(5000)$ L is the sample number at which this maximum occurs
6: $x_s = x(0 : 5L)$ ▷ extract a segment from $x(0)$ to $x(5L)$ into x_s
7: $e = E(0 : 5L)$ ▷ extract a segment from $E(0)$ to $E(5L)$ into e
8: **end procedure**

4.5 Feature Extraction

The first step in feature extraction was to extract three features from segments of envelope signals:

1. The number of peaks higher than 0.1 that a segment contains.
2. The average distance between adjacent peaks.
3. The energy of the segment.

These features are aimed primarily at separating murmur and murmurless heart sounds. The rationale behind feature 1 and 2 is that murmurs tend to generate many extra peaks that are close together in the envelope signal, and the reason behind feature 3 is that murmurs increases the overall energy of heart sounds. It can be seen that these features contain no

time information, that is, it does not matter where the peaks occurs. This means that the alternative to segmentation discussed in the previous section is justified, since with these features it is only necessary to extract the same number of cardiac cycles. The cardiac cycles do not even have to be whole, a segment can begin with half a cycle, then four whole cycles and end with another half cycle.

4.5.1 Peak Detection

Peak detection used in this work was more elaborate than simple thresholding. Any maxima in the envelope signal that is higher than the "valley" on both sides by a certain threshold was considered a peak. The pseudo code for this peak detection algorithm is shown in algorithm 4.3 In order to clarify the difference between the peak detection in algorithm

Algorithm 4.3 Peak detection algorithm

1: **procedure** PEAKDET(E, Δ) ▷ vector E contains envelope signal whose peaks to be detected and Δ is the threshold, a maxima must be proceeded to the left by a value lower than it by Δ
2: max = blank 1 by 2 matrix
3: $x = 1, 2, ..., \text{length}(E)$
4: $m_x = -\infty$
5: $mxpos = 0$
6: **for** $i = 1, \text{length}(E)$ **do**
7: $this = 1$
8: **if** $this > m_x$ **then**
9: $m_x = this$
10: $mxpos = x(i)$
11: **end if**
12: **if** $this < m_x - \Delta$ **then**
13: $max = \begin{bmatrix} max \\ mxpos & x \end{bmatrix}$ ▷ concatenate the row vector $[mxpos \quad x]$ vertically with matrix max
14: **end if**
15: **end for**
16: **end procedure**

4.3 and a simple thresholding peak detection method, consider Figure 4.4 showing one cardiac cycle of the envelope signal of a heart sound with aortic stenosis. The line indicate the threshold value of 0.2. If simple thresholding peak detection was used in the situation in Figure 4.4, it would pick up only three peaks, since there are 6 threshold crossings as indicated by the circles. On the other hand, peak detection using algorithm 4.3 would pick up

any maxima that is monotonically decreasing to its left to a value lower than the maxima by Δ If Δ is equal to 0.2, the peak detection algorithm 4.3 would detect 12 peaks. The reason

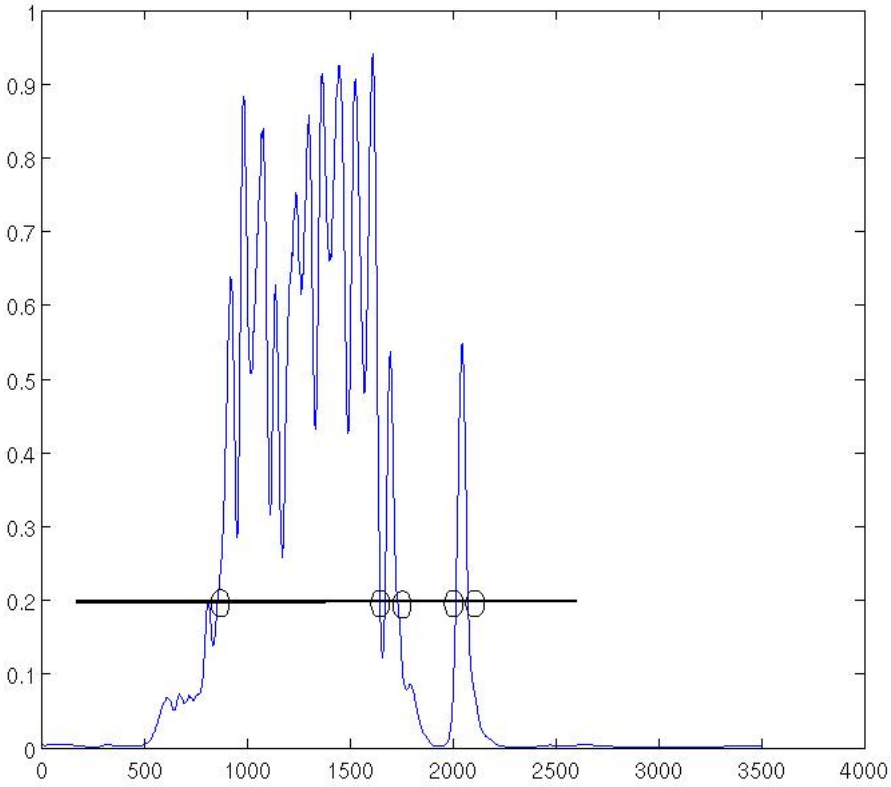


Figure 4.4: Illustration of difference between simple peak detection and algorithm 4.3

why using algorithm 4.3 is an advantage over simple thresholding can be seen by examining the envelope (5 cardiac cycles) of the heart sound with aortic stenosis shown in Figure 4.5 If the threshold was set at 0.1, simple thresholding would detect 10 peaks, which would be the same number of peaks that would be detected from a normal heart sound by simple thresholding because 5 cardiac cycles of normal heart sound would have 10 peaks, two (S1 and S2) for each cardiac cycle. On the other hand, the peak detecting scheme used in this work with $\Delta = 0.1$ would detect many more peaks, somewhere around 60 for the situation in Figure 4.4. Moreover, the average distant between adjacent peaks would also be different from that obtained by simple thresholding since many more peaks are detected. It is desirable for normal and abnormal heart sound to have significantly different feature values. Therefore, peak detection method in algorithm 4.3 is preferred over simple thresholding.

4.5.2 DWT Coefficients Features

In the second feature extraction step, 32 features were extracted from segments of heart sound themselves. The feature extraction scheme for this step is a modification to the method

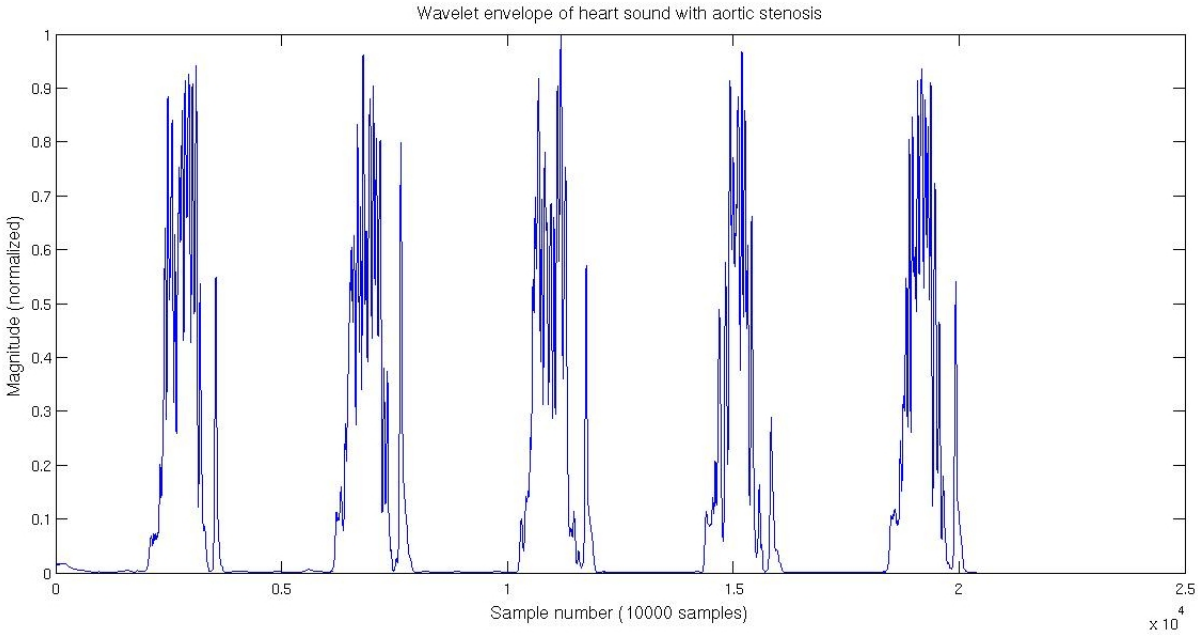


Figure 4.5: Envelope signal of heart sound with aortic stenosis

in [8].

Each object was decomposed using DWT with Daubechies-2 mother wavelet. The 5th level details coefficients all through the 3rd details coefficients were concatenated together to form a new signal. This new signal was then divided into 32 non-overlapping windows and the signal energy was calculated for each window to form each element of the 32-elements feature vector.

In [8], each cardiac cycle was segmented precisely. That is, the start of each segment coincides with the start of a cardiac cycle. In this work however, feature extraction was performed on the segments of heart sound with length equal to five cardiac cycle where the start of the segment does not necessarily coincide with the start of the first cardiac cycle in the segment. However, the offset between the start of the first cardiac cycle and the start of the segment is small compared to the total length of the segment that its affects on the DWT coefficient is insignificant. This is justified by the result of the ten-fold cross validation. The 3-elements and 32 elements feature vectors are concatenated together to form 35-elements feature vector for each heart sound in the training set. Each feature variable was then normalized by subtracting and dividing by their respective means and standard deviations. The whole feature extraction process is described in algorithm 4.4

4.6 PCA Analysis

Principle component analysis (PCA) technique is used in statistical pattern recognition in order to two accomplish two tasks [5]. The first is to maximize the variance of each feature

Algorithm 4.4 Feature extraction

procedure FEATUREEXTRACTION(x_s, e) ▷ x_s and e have the same meaning as from algorithm 4.2
  $p = \text{peakdet}(e, 0.1)$ ▷ the function peakdet is from algorithm 4.3
  $f_1 = \text{length}(p)$ ▷ first feature is the number of peaks
  $f_2 = \text{mean}\, p_{2,3,..,n} - p_{1,2,...,n-1}$ ▷ second feature is the average distance between adjacent peaks
  $f_3 = \sum_{i=0}^{N-1} e(i)$ ▷ third feature is sum of all samples of e
  $d_2 = \text{DWT}(x_s, 6, \text{Deubechies-2}, \text{details-2})$ ▷ d_2 is the DWT coefficient at details level 2 using Deubechies-2 wavelet with 6 decomposition levels
  $w = \text{floor}(\text{length}(d_2)/32)$ ▷ divide d_2 into 32 non-overlapping window
  $j = 1$
  for $i = 1 \rightarrow 32$ **do**
    $F(i) = \frac{1}{w} \sum_{j}^{j+w-1} d_2(j)$
  end for
  $F = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ F \end{bmatrix}$ ▷ F is the final feature vector
end procedure

variable by projecting them onto the direction of maximum variance. Maximizing variance makes the classifier easier as shown in Chapter two. The second is to reduce the dimension of the features space by dropping eigenvectors that correspond to small eigenvalues from Equation 4.6. An element of a new feature vector can be formed by,

$$x'_i = \mathbf{w_i}^{\mathbf{T}}\mathbf{x} \quad \forall \mathbf{x} \in \mathbf{X}, \mathbf{i} \in \{\mathbf{1}, \mathbf{2}, ..., \mathbf{m}\}, \tag{4.6}$$

where x'_i is an i$^{\text{th}}$ element of a new feature vector, $\mathbf{x}$ is the original feature vector of an observation, and $\mathbf{w_i}$ are eigenvectors with m largest eigenvalues. The eigenvector and their eigenvalues are calculated from the covariance matrix estimated from the original training set $\mathbf{X}$. An unbiased estimate of the covariance matrix $\mathbf{C}$ given a set of vectors is given by,

$$\mathbf{C} = \frac{\mathbf{1}}{\mathbf{N-1}} \sum_{\mathbf{i=1}}^{\mathbf{N}} (\mathbf{x_i} - \overline{\mathbf{x}})(\mathbf{x_i} - \overline{\mathbf{x}})^{\mathbf{T}}, \tag{4.7}$$

where N is the number of data points in the training set (57 in this case), $\mathbf{x_i}$ is a feature vector and $\overline{\mathbf{x}}$ is the mean vector. Each new feature value x_i is a linear combination of the elements of the feature vector $\mathbf{x}$. The dimension of the feature vectors can be reduced by choosing $i \in 1, 2, ...m < n$, where n is the dimension of the original feature vectors and m is the number of significant eigenvalues. Usually $n > m$ since many eigenvalues of C are small. A pseudo code for PCA is given in Algorithm 4.5.

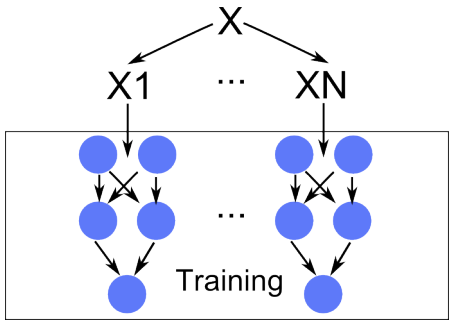
Algorithm 4.5 PCA analysis

procedure PCA($\mathbf{X}$)
  C= 38 by 38 matrix of zeros
  N =number of columns of $\mathbf{X}$
  for $i = 1 \rightarrow N$ **do**
    $\mathbf{C} = \mathbf{C} + (\mathbf{x_i} - \overline{\mathbf{x}})(\mathbf{x_i} - \overline{\mathbf{x}})^{\mathbf{T}}$
  end for
  $\mathbf{C} = \frac{\mathbf{C}}{\mathbf{N-1}}$
  calculate eigenvalues and eigenvectors of C
  Let $E = [\mathbf{v_1^T}, ..., \mathbf{v_N^T}]$ where each row of E is an eigenvector and
  Let $D = [d_1, ..., d_N]$ their eigenvalues, where $d_i \geq d_{i+1}$
  S =sum of D
  $k = Q = 0$
  while $Q \leq 0.9S$ **do**
    $Q = Q + D(k)$
    $k = k + 1$
  end while
  $E = E(k+1, ..., N)$ ▷ drop the $k+1^{\text{th}}$ till N^{th} rows of E
  for $i = 1 \rightarrow N$ **do**
    $\mathbf{y_i} = \mathbf{E}\mathbf{x_i}$ ▷ Each $\mathbf{y_i}$ is a new feature vector
  end for
end procedure

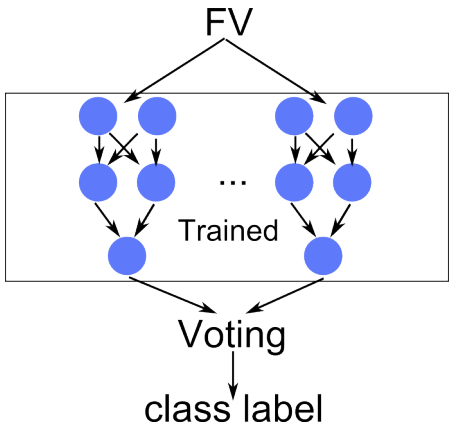
4.7 Bagging Classifiers

In complex pattern recognition tasks, classification performance can be improved by using *bagging* method which uses multiple classifiers instead of just one. Bagging is a two steps process. First step is called *bootstrapping* where the original training set with Q samples $\mathbf{X} = \{(\mathbf{x_k}, \mathbf{d_k})\}_{\mathbf{k=1}}^{\mathbf{Q}}$ is randomly sampled such that M new training sets, $\mathbf{X_1}, \mathbf{X_2}, ..., \mathbf{X_M}$, each with Q samples are obtained. Some samples from $\mathbf{X}$ may be missing and some samples may be repeated for each of the $\mathbf{X_i}$. This is done so that each classifier is trained with different training data, otherwise, each of them would be the same and there would be no point in having multiple classifiers. Bootstrapping is a common practice in statistics to deal with small data set, where more data is expensive or difficult to collect.

In the second step of bagging, M classifiers are trained, each with its own training set $\mathbf{X_i}$. The final decision of an object's class is obtain by voting each of the individual classifier's output. Figure 4.6 shows that illustration of bagging using neural network as classifier. In Figure 4.6(a), M neural network are being trained with different training set generated by bootstrapping, and in Figure 4.6(b), the neural network's make their collective decision by voting (Equation 4.9) with their individual outputs. It was shown in [26] that the per-



(a) Training of bagging predictor



(b) Making decision by voting

Figure 4.6: Illustration of bagging

formance of bagging classifiers is equal to or greater than the performance of any single classifier, provided that each of the individual classifier trained bagging are themselves reasonable classifier. In this work, M was chosen to be 10, because the result in [26] showed that there is little improvement when M is made larger. Additionally, because of the imbalance between normal and abnormal heart sounds in the training set, each of the training set created by bootstrapping was such that there are equal number of the two classes in each of the training set. This is necessary because classifiers trained with imbalanced training set tend to favor the majority class, as described in [26].

4.8 Ten-folds Cross-validation

Cross-validation is a technique used in statistical pattern recognition to assess the performance of a classifier on an independent (not part of the training set) data. One pass of cross-validation involves partitioning the training data into complementary subsets, training the classifier using the larger subset and then test the performance of the trained classifier using the smaller subset. One pass of cross-validation is called a fold. This is illustrated in Figure 4.7 for the five folds case. The whole training data is divided into five portions. In each fold, the green subset (4 portions) is the training data, while the blue subset (1 portion)

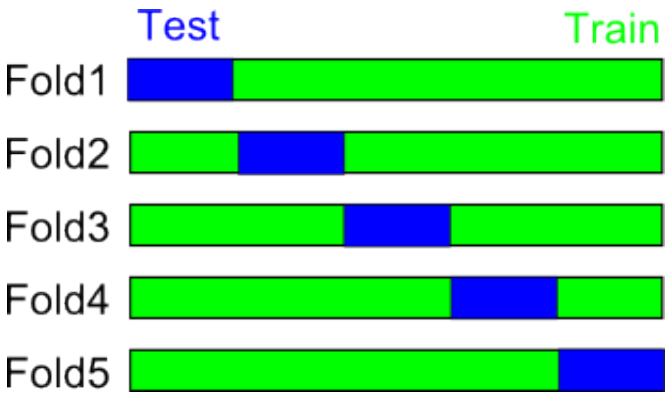


Figure 4.7: Illustration of cross validation with five folds

is reserved for testing, that is, the classifier had never seen the blue subset prior to testing phase. As the fold advances, each portion takes turn being the test subset. Once all five folds had been completed, all data had been tested on the classifier as unseen samples.

The standard ten-folds cross-validation was used in this work. In each fold, 10 $\%$ of the data had been tested and the outputs due to those samples are stored. At the end of cross validation, the same number of outputs as the number of samples in the training set was obtained, each tested as unseen samples. The specific ten-fold cross validation process that incorporate bagging classifiers used in this research is shown in Figure 4.11, at the end of this Chapter due to the Figure's size. This Figure requires some explanation: first the training data is divided into ten portions, then in each fold the training portions are sampled according to bootstrapping procedure. This generates M different training portions for each of the M neural networks. Each neural network is then trained using its unique training portion and tested on the test portion, **which is the same for all neural networks** since this portion was not sampled. This repeats 30 times and best (now a neural network is best is discussed below) neural networks were kept. At the end of the fold the outputs of the M best neural networks over the test portion enter voting, which generates the class labels for the test portion. This completes one fold of cross-validation and the whole procedure is repeated again using different fold as test portion until all the data had been used as test portion.

From Chapter two it was shown that an neural network's performance is dependent upon its initial random weights because the error function may be stuck in a local minimum during the training process. Thus in each cross-validation fold, the neural network is trained, initialized and retrained 30 times, and each time its performance was evaluated. The best performing network and its outputs due to the test portion of the training data for that particular fold were kept. An neural network is considered best if its sum of absolute error (SAE) over the test portion is minimum. The SAE is is calculated by taking the difference between the network's actual output and the desired class label, take the absolute value and summing over the test portion. The value of SAE that is minimum indicates that the neural network that produces it has the best performance, that is, for each time the neural network is trained,

$$\text{SAE} = \sum_{\forall i \in T} |y_{ki} - d_i|, \tag{4.8}$$

where T is the test portion of the training data for that fold, y_i is the actual output of the neural network for input x_i and d_i is the desired output (0 for one class and 1 for another). Before each cross-validation fold, a variable "minSAE" is initialized to some arbitrary large number, and when the current neural network's SAE is less than minSAE, minSAE is replaced by SAE and the neural network itself is saved as the object "bestNN", this continued until the neural network had been trained 30 times, after which "bestNN" holds the best performing neural network for that particular cross-validation fold. And in each fold, there are M processes of training and retraining of the neural networks happening concurrently, represented by the bounding boxes in Figure 4.11

4.8.1 Making Decision by Voting

At the end of each round of cross-validation, after each of the M neural network had been trained and retrained 30 times and the best ones according to Equation 4.8 had been identified. The neural networks make their collective decision on the type of a heart sound by Equation 4.9

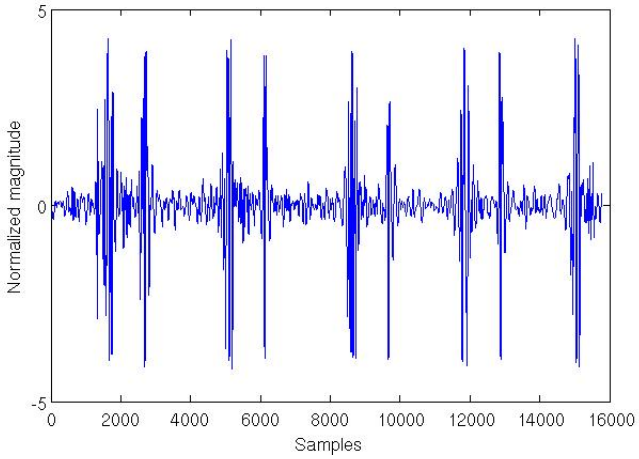
$$c = \begin{cases} 1 & \text{if } \left(\sum_{i=1}^{M} \text{round}(y_i)\right) \geq T \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

where y_i is the output of the i^{th} neural network, M is the number of classifiers, T is the threshold value and c is the class label. Ordinarily, the threshold is at 5 (halfway between 0 and 10), but it can be adjusted to take into account the cost of each type of miss-classification. That is, it is desirable to allow some false positive (healthy sounds classified as abnormal) in order to minimize the false negative (abnormal sound classified as healthy) because the latter case is much more costly on the patient. Thus the threshold should be set closer to 0 than to 10. The value of this threshold is evaluated in Chapter 5.

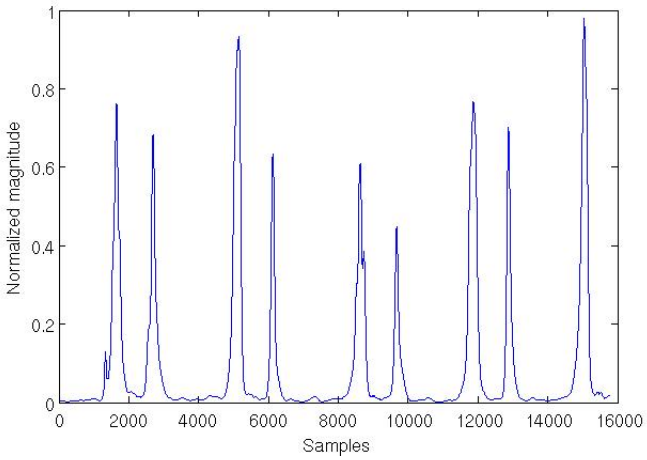
4.9 Analysis of the proposed algorithm

In this section, how the algorithm works is clarified. Figure 4.8 (a) and (b) shows a segment with length equal to five cardiac cycles of a heart sound signal and its envelope, respectively. These segments are obtained from the proposed segmentation scheme, their lengths are about five cardiac cycle, but they may not necessarily contain five *complete* cardiac cycles. That is, there can be discrepancy between the start of the first cardiac cycle in the segments and the start of the segments themselves, shown in Figure 4.9 by the double head arrow. Subsequently this distant is referred to as offset. The offset is not constant because

the segmentation scheme randomly crop out five cardiac cycles piece from anywhere within the signals. This raises a question whether or not feature extracted from these segment are valid. The goal of the discussion in this section is to demonstrate the validity of the proposed heart sound analysis method. Consider the first three elements of the feature vectors, which



(a) a



(b) b

Figure 4.8: Heart sound and envelope segments obtained by proposed segmentation method

are obtained from the envelope signal; the number of peaks, the average distant between adjacent peaks and the energy of the envelope signal - these features contain no temporal information. That is, the information needed to calculate these three features are provided by the peak detection algorithm, even for the second feature where the distant (hence timing information) between adjacent peaks are concerned, but these values get averaged through the whole signal. This means that the offset does not have any effect on the values of the first three features, as long as the envelope signals have lengths equal to same number of cardiac cycles. Figure 4.10 shows a comparison between the envelope signal of normal, S3 and AS

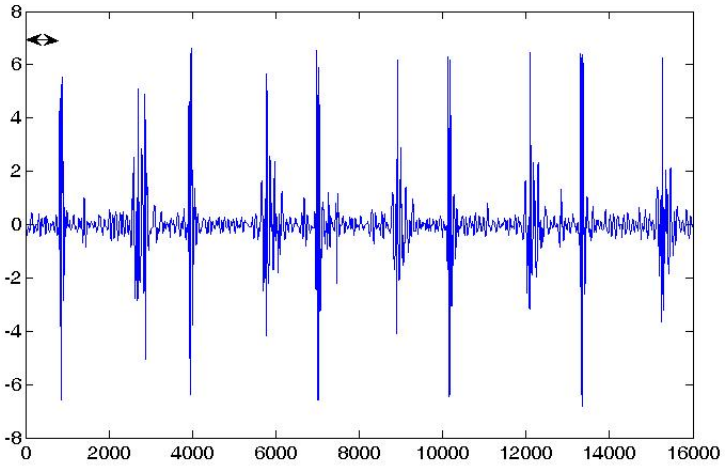


Figure 4.9: Offset between the start of first cardiac cycle and start of segment

heart sound respective from top to bottom panel. It can be seen that the number of peaks are greater, the peaks are closer together and the envelope signal energy is higher for abnormal heart sounds. For the remaining 32 elements of the feature vectors that are obtained using

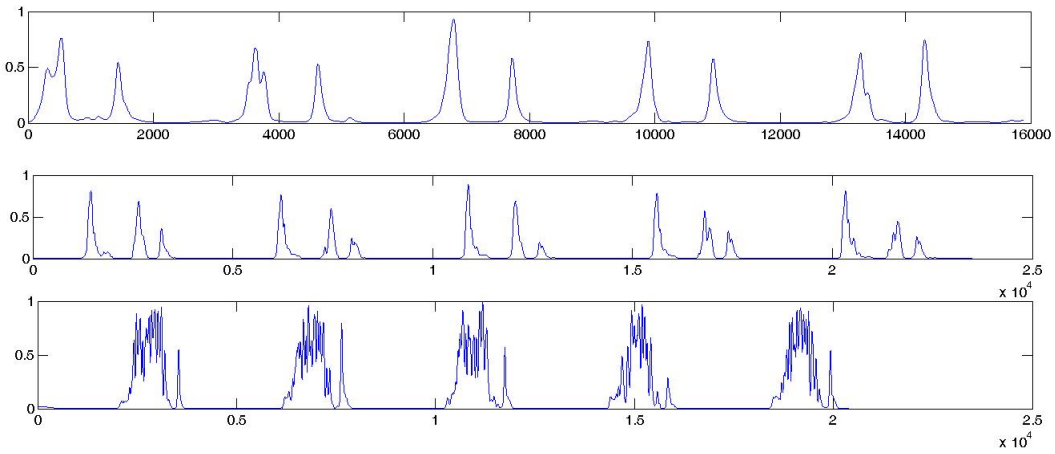


Figure 4.10: Envelope signal of normal (top), S3 (middle) and AS (bottom) heart sound

the method in [8]. Again the concern is about the offset, which can potentially change the value of the d2 DWT coefficients. The DWT is a time-frequency representation of signals, therefore this offset could potentially have some effect on the coefficients. However, it can be seen in Figure 4.9 that the length of the offset is small compared to the length of the segment. The effect of the offset can be thought of as ”shifting” of the d2 coefficients to the right by the same amount as the length of the offset. The maximum offset length is just under the length of one cardiac cycle, since shifting exactly one cardiac cycle is equal to no shifting

at all. Assuming that the distribution for the offset is uniform, the expected offset is half the cardiac cycle length. At a typical heart rate of 70 bmp, one cardiac cycle is roughly equal to 4700 samples. The length of the entire segment is thus $4700x5 = 23500$, which means that the length of a window is $23500/32 = 734$ samples. Half a cardiac cycle is 2350 samples, which is equal to about 3 widows. Since the heart sound is periodic, the effect of the offset is the same as if one begin with zero offset, calculate the feature vector and "circular shift" it by 3. Circular shifting by just 3 elements in a vector that is 32 elements long, whose adjacent elements must have similar values due to the averaging effect of calculating signal power, will end up in a vector this not too far from the original since the norm remains the same and only the direction slightly changed. It can be concluded that the effect of the offset on the feature vectors is small, and even made smaller by the PCA process that follows after feature extraction.

4.9.1 Interpretation of the output of neural network

From Chapter two it was explained that pattern recognition problem can be formulated as selecting the class label c_i that maximizes the posterior probability $p(c_i|\mathbf{x})$ given the feature vector $\mathbf{x}$. When there is only one feature this can easily be accomplished using Bayes' theorem. However, when the number of feature is large, it become necessary to estimate multi-variables class-conditional pdf's $p(\mathbf{x}|\mathbf{c_i})$, which require large training data for high dimensional feature vectors (the number of data points needed is proportional to the square of the dimension of the feature vector, so in the case of this work with 8 dimensional feature vector, almost 2000 samples is needed). This is clearly impractical and to handle the high-dimensional problem, neural network are used, and it can be shown that the output of a neural network can be regarded as posterior probability. For the two-classes problem such as the one in this work, there is only a single output neuron whose output can be regarded as $p(c_1|\mathbf{x})$, which is the probability of the observed feature vector belonging to class 1. Since there are only two classes, the probability of NOT being in class 1 is simply $1 - p(c_1|\mathbf{x})$. This means that one should assign class 1 if the output of the neural network is greater than or equal to 0.5, and assign class 2 otherwise. The proof that neural network output can be regarded as posterior probability can be found in Chapter 7 of [10]. Here only a brief summary is included. Suppose that the training set $T = (\mathbf{x_k}, \mathbf{d_k})_{k=1}^{Q}$ with Q members where $\mathbf{x_k}$ is the k feature vector and d_k is its corresponding desired response, sampled from a joint probability function $p(\mathbf{x_k}, \mathbf{d_k})$. The likelihood of T is given by

$$L = \prod_{k=1}^{Q} p(\mathbf{x_k}, \mathbf{d_k}) \tag{4.10}$$

taking the negative log of this and assuming that the neural network's output is deterministic with additive unit Gaussian noise lead to the sum of squared error function

$$E = \frac{1}{2} \sum_{k=1}^{Q} (d_k - f(\mathbf{x_k}, \mathbf{w}))^{\mathbf{2}} \tag{4.11}$$

where f is a vector-valued function that take the feature vector $\mathbf{x}$ and weight vector $\mathbf{w}$ as its input. This function represent the neural network itself. Letting $Q \to \infty$, a lengthly manipulation can show that

$$E = \frac{1}{2} \int (E[d|\mathbf{x}] - \mathbf{f}(\mathbf{x}, \mathbf{w}))^{\mathbf{2}} \mathbf{p}(\mathbf{x}) \mathbf{dx} \tag{4.12}$$

which is minimized when $f(\mathbf{x}, \mathbf{w}) = \mathbf{E}[\mathbf{d}|\mathbf{x}]$, which is the expected value of $p(d|\mathbf{x})$.

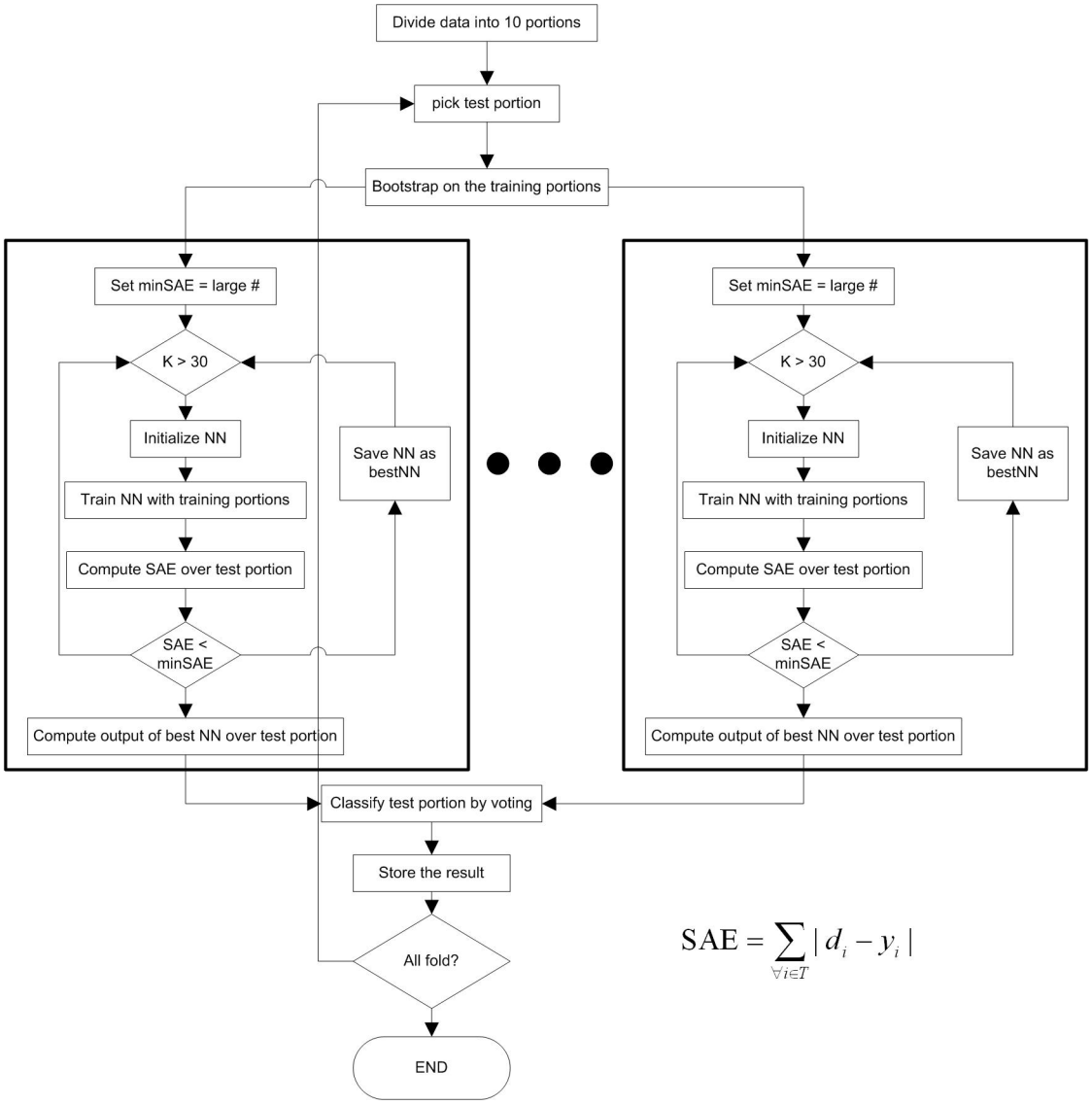


Figure 4.11: Ten-folds cross-validation with bagging neural network classifier flowchart

Chapter 5

Experimental Results

This chapter shows the result of applying the proposed heart sound analysis algorithms discussed in the last chapter to the training set shown in Table 4.1. The training data had been modified by copying the normal heart sounds so that there roughly the same number of samples for each class. This was done to prevent the classifier(s) from favoring the majority class. Thus the training set $\mathbf{X}$ now consists of **97 feature vectors, 48 for normal and 45 for abnormal class**. The results are separated into three experiments, the first is cross-validation using a single neural network classifier with non-sampled training data $\mathbf{X}$. The number of hidden neurons was varied to determine the optimal number of hidden neuron. The second experiment was cross-validation using bagging classifiers and number of hidden neuron from obtained from experiment 1 and the number of bagging classifiers was varied. Finally in experiment 3, the number hidden neuron and the number of bagging classifiers were fixed using the results of the first two experiments and the decision threshold (T in Equation 4.9) was varied.

5.1 Experiment 1: Cross-validation using Single Neural Network

As explained in Chapter 2, classification results can be demonstrated by a set of three numbers: accuracy, sensitivity and specificity. These numbers in turn are calculated from a set of 4 numbers that provide the ”raw score” of a pattern recognition system. They are true positive (TP), false positive (FP), true negative (TN), and false negative (FN). They are defined as follows:

- TP: the ratio between samples which are actually positive over the number of samples classified as positive.
- FP: the ratio between samples which are actually negative over the number of samples classified as positive.
- TN: the ratio between samples which are actually negative over the number of samples classified as negative.
- FN: the ratio between samples which are actually positive over the number of samples classified as negative.

And accuracy, sensitivity and specificity are defined as the following,

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{N} \tag{5.1}$$

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.2}$$

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{5.3}$$

where N is the number of samples in the training set. The result of experiment 1 in shown in Table 5.1 where N denotes the number of hidden neurons. The result in Table 5.1 shows

Table 5.1: Varying Number of Hidden Neurons

N	**TP**	**TN**	**FP**	**FN**	**Acc.**	**Sen.**	**Spec.**
1	36	48	0	9	90.3	80	100
2	36	48	0	9	90.3	80	100
3	36	48	0	9	90.3	80	100
5	38	48	0	7	92.5	84.4	100
7	38	48	0	7	92.5	84.4	100
10	38	48	0	7	92.5	84.4	100
20	37	48	0	8	91.4	82.2	100

that the best performance of a single neural network occurs at 5 hidden neurons. However, considering that the performance of is not much lower (just over 2 %) for the case of one hidden neuron, it raises a question whether or not $\mathbf{X}$ is linearly separable. This is because if the hidden layer consists of only a single neuron the output neuron is just an identify map and can be ignored and the neural network is reduced to just two layers. A classification problem is said to be linearly separable if for a training set in R^n , there exist a hyperplane in R^{n-1} that can divide the space R^n, into two mutually exclusive regions each containing exclusively members of one class. A simple way to check whether or not $\mathbf{X}$ is indeed linearly separable, is a test using a simple perceptron model (a single neuron with a binary threshold function as activation function) that is,

$$y = \begin{cases} 1 & \text{if } \mathbf{wx} \geq \mathbf{0} \\ 0 & \text{otherwise}, \end{cases} \tag{5.4}$$

where y is the output of the perceptron, $\mathbf{x}$ is the input vector and $\mathbf{w}$ is the weight vector. The perceptron is the simplest neural network model that can only handle linearly separable classification problem. If the perceptron is successful, then $\mathbf{X}$ is linearly separable, otherwise, it is not. The perceptron was trained using $\mathbf{X}$ and also tested using $\mathbf{X}$ to give it the best possible chance. The result of perceptron trails is given in Table 5.2 The result in Table

Table 5.2: Perceptron Test

Trial	TP	TN	FP	FN	Acc.	Sen.	Spec.
1	24	48	0	21	77.4	53.3	100
2	39	20	28	6	63.4	86.7	41.7
3	40	36	12	4	81.7	90.1	75.0
4	40	28	20	5	73.1	88.9	58.3
5	36	36	12	9	77.4	80.0	75.0
6	25	48	0	20	78.5	55.6	100
7	41	28	20	4	74.2	91.1	58.3
8	37	44	4	8	87.1	82.2	91.7

5.2 shows that the perceptron fails to classify $\mathbf{X}$. In trial number 6, all the negative samples were correctly classified with about half of the positive samples were miss-classified. And in trial 7, most of the positive samples were correctly classified while about half of the negative samples were miss-classified. This shows that there are overlap of the data points such that trying to correctly classify one class will result in another being significantly miss-classified. Thus it can be concluded that $\mathbf{X}$ is NOT linear separable.

5.2 Experiment 2: Varying the Number of Classifier

In experiment 2, the cross-validation result using different number of bagging classifiers were tested. The number of hidden neuron was fixed at 5, which yielded maximum performance in experiment 1. The decision boundary is the median value. That is, if the number of classifiers is 4, then the decision threshold is 2 since all possible output are $\{0, 1, 2, 3, 4\}$. Therefore the number of classifier must be even because zero is also counted, so the number of possible output is odd and the median can be selected as the decision threshold. Table 5.3 shows the result of varying the of bagging classifiers. The result shows that there is little improvement as the number of classifiers is increased, so it could be set to a low number to save on the processing time.

5.3 Experiment 3: Varying the Number of Classifier and Decision Threshold

For experiment 3, the decision threshold is varied, and doing so may have some effect on the results of experiment 2 as well. In other words, experiment 2 and 3 may not be independent of each other, which means that to find the optimal values for number of classifiers and decision threshold, all possible combinations must be enumerated. This is done by following the same steps as in experiment 2, but for each number of classifier vary the decision threshold such that it include all values from the median down to 1 (the threshold was

Table 5.3: Varying Number of Classifiers Test

Num. Class.	TP	TN	FP	FN	Acc.	Sen.	Spec.
2	36	48	0	9	90.3	80.0	100
4	38	48	0	7	92.5	84.4	100
6	38	48	0	7	92.5	84.4	100
8	39	48	0	6	93.4	86.7	100
10	37	48	0	8	91.4	82.2	100

not increased since in all cases so far the specificity is already 100, increasing the decision threshold would only make the performance worse). The result is shown in Table 5.4, where the left column titled "Combination" has the following format: 4.1 means four classifiers with decision threshold equal to 1, 6.2 means six classifiers with decision threshold equal to 2, etc. The case where number of classifiers equal to two was omitted since there is only one possible decision threshold. The result shows maximum performance at 6.1 and 8.1 (six and eight classifier respectively with decision threshold equal to 1), and 6.1 is more desirable since it runs faster. From experiment 1 through 3, it can be concluded that the optimal configuration for the bagging classifier are: **5 hidden neurons, 6 classifiers, and decision threshold equal to 1**.

Table 5.4: Varying both Number of Classifiers and Decision Threshold Test

Combination	TP	TN	FP	FN	Acc.	Sen.	Spec.
4.1	39	48	0	6	93.5	86.7	100
4.2	38	48	0	7	92.5	84.4	100
6.1	41	48	0	4	95.7	91.1	100
6.2	37	48	0	8	87.1	82.2	100
6.3	38	48	0	7	92.5	84.4	100
8.1	41	48	0	4	95.7	91.1	100
8.2	39	48	0	6	93.5	86.7	100
8.3	39	48	0	6	93.5	86.7	100
8.4	39	48	0	6	93.5	86.7	100

Chapter 6

Conclusion

A complete procedure for heart sound analysis was developed based on the pattern recognition approach and new approach to segmentation of heart sound was proposed. The proposed segmentation method does not require the type of each FHS to be identified, is not based on the assumption that systole is shorter than diastole that is usually made in many studies, and is robust to all heart sound types: from normal to severe murmur. This study also apply PCA analysis and bagging classifier technique to improve the classification performance. Experiments was conducted to determine the optimal configuration for the classifier. This optimal configuration yield a performance of 95.7% accuracy, 91.1% sensitivity and 100% specificity. Given this level of performance it is reasonable to say that the proposed method is ready to be tested on real patient in an actual hospital. The advantages of this work are:

1. The segmentation method make no assumption about the cardiac cycles and is robust to different type of heart sounds
2. There is no need to perform complicated envelope analysis of heart sound signal
3. The proposed method achieved good performance for a large variety of heart sound types.

And the shortcomings of this work are:

1. Only classifier heart sounds as normal or abnormal, with no diagnosis about the type of the disease (multi-class classification).
2. The training set is relatively small
3. The cross-validation time is quite long, around 250 to 300 seconds.

6.1 Comparison with Other Studies

Comparison among different studies in the field of heart sound analysis is difficult, mainly because there is no ”standard heart sound set” on which different algorithms can be tested fairly. Also, the way of reporting the performance can be different, some studies report result based on the number of heart sounds, while some report result based on number of feature vectors (often one feature vector is extracted from each cardiac cycle, so there can be more than one feature vector per heart sound). Many studies rely one manual segmentation or segmentation using the ECG signal, thus quoting classification accuracy alone would be

unfair for those that use automatic segmentation as in this work. Considering all this difficulty, this work is compared to the study in [16] that is quite similar to it. In that work heart sounds were classified into three types: normal, systolic murmur and diastolic murmur, they reported an average classification accuracy over three different datasets of 97%. This number of was based feature vector unit, not heart sounds unit and the study did not incorporate abnormal heart sound there are non-murmur types (S3, S4, systolic click, etc). This shows that the proposed method in this work is on par with results reported in the literature, while being applicable all types of abnormal heart sounds.

6.2 Further Work

The work in this research up to this point had provided a proof of concept for the proposed heart sound analysis method. All of the software are currently in MATLAB code (see appendix A). Further work can be divided into two parts: practical and research. For practical aspect, further work include implementing the software so that it could be run independently out of the MATLAB environment and adding a GUI to make the system more user friendly. This is so that the system could be tested on live patients in real hospital environment to verify its effectiveness in real-life situation.

For research aspect there are several possible directions. In a two classes problem considered in this work, it is possible to solve such problem using "single class classification" approach, where the training data consists of samples from only a single class. In the case of heart sound analysis, this would mean collecting only samples of normal heart sounds, then perform segmentation and feature extraction as usual. Once the set of training vectors $\mathbf{X}$ is obtained, a hypersphere that tightly encloses the training vectors is defined. Any new feature vector that lies inside this hypersphere is considered normal, while those that lie outside is considered abnormal. Using this approach there would be no need to collect data from ill patients, which is difficult to obtain in large quantity. Another possible direction is to borrow from speech recognition, using linear predictive coding together with sliding windows to extract feature vectors without any segmentation of heart sound.

Appendix A

MATLAB Source Codes

Listing A.1: Preprocessing

```
%Preprocessing
function [x fs] = preprocessing(hs,Fs)

%change sampling rate by rational factor so final sampling rate is at 4000
if Fs == 8000
    x = decimate(hs,2,'fir');
    fs = Fs/2;
elseif Fs == 8012
    x = decimate(hs,2,'fir');
    fs = Fs/2;
elseif Fs == 11025 %downsample by 0.36
    %x = decimate(hs,3,'fir');
    x = resample(hs,160,441);
    fs = 4000;
else
    %x = decimate(hs,10,'fir');
    x = resample(hs,40,441);
    fs = 4000;
end

%load BPF;
%x = filter(Hbp,x);

%noise removal using wden, db6 wavelet, 5 level DWT
x = wden(x,'heursure','s','mln',5,'db6');
%normalization
x = (x-mean(x))/std(x);
end
```

Listing A.2: Envelope detection and cycle length calculation

```
%envelope and cycle length detection
function [E L] = envelopeDetection(x, fs)
    scales = [20 22 24 25 28 30 33 37 41 44 47 51 55 60 66 73 82 88 94....
        101 110 120 132 147 165 176 189 204 220 241 265 294 330 353....
        378 407 441 481 529 588 661];
    Y = cwt(x,scales,'cmor1-1');
    E = abs(Y).^2;
    E = sum(E);
    E = E/max(E);
    r = xcorr(E);
    r = r(ceil(length(r)/2):length(r));
    [M L] = max(r(1000:5000)); %start seaching 1000 samples from the beginning of
    %the signal to get pass the peak at origin of r
```

```
14      %must add 1000 to the index because we're searching a shorter vector by starting
15      %at 1000 samples from beginning, and the index value will be wrong
16      %otherwise. And limit the search till only th 5000th sample because the
17      %r signal may be higher peak later
18      L = L+1000;
19  end
```

Listing A.3: Feature extraction

```
1   function [f1 f2] = featureExtraction(env,segment)
2   %feature 1 = number of peaks
3   peaks = peakdet(env,0.1);
4   feature1 = length(peaks(:,1));
5   %feature 2 = average distance between peaks
6   tmp1 = peaks(:,1);
7   tmp1(1) = [];
8   tmp2 = peaks(:,1);
9   tmp2(length(peaks(:,1))) = [];
10  distance = tmp1-tmp2;
11  feature2 = mean(distance);
12  %feature 3 = area under the envelope
13  feature3 = sum(env);
14
15  f1 = [feature1;feature2;feature3];
16
17  [C,L] = wavedec(segment,6,'db2');
18  d2 = C(sum(L(1:5)):sum(L(1:6)));
19  samples_per_window = floor(length(d2)/32);
20  f2 = zeros(32,1);
21  j = 1;
22  for i = 1:32
23      f2(i) = (1/samples_per_window)*abs(sum(d2(j:j+samples_per_window-1)))^2;
24      j = j + samples_per_window;
25  end
```

Listing A.4: Bootstrapping

```
1   function [bagging]=bootstrap(FS,label,M)
2   %BOOTSTRAP perform the bootstrap on set of feature vector "FS"
3   %   the feature vectors are arranged on the columns of "FS" so that
4   %   each column is an observation and each row is a feature variable
5   %   "label" is a row vector of class label and "M" is the number of
6   %   new feature vector sets that we would like to generate
7
8   [row col]=size(FS);
9   labeledFS = vertcat(FS,label);
10  bagging=zeros(row+1,col,M);
11  for i=1:M
12      r=randi(col,[1,col],'uint8');
13      bagging(:,:,i) = labeledFS(:,r);
14  end
```

Listing A.5: Cross validation

```
1   function [testResult]=finalTest(X,label)
2   %finalTest performs ten fold cross validation using committee of NN
3   %   "X" is a matrix with PCAed feature vectors as columns, label is a row
4   %   vector of class label. It works like this:
5   %   1. divide X into ten portions using crossvalind
```

```
6  %    2. each fold, the training portions is bootstraped
7  %    3. each feature vector set generated with bootstrapping used to train a
8  %       NN minimizing the SAE
9  %    4. trained committee is used to classify the testing set
10 %    5. repeat for all folds
11
12 M = 6;
13 [row col] = size(X);
14 hiddenNeurons = 5;
15 numIteration = 10;
16 indices = crossvalind('Kfold', col, 10);
17 nnComittee = cell(M,1);
18 %T = 0.5;
19 testResult(1,1:col) = 0;
20 for i = 1:10
21     testIndex = (indices == i); trainIndex = ~testIndex;
22     bootstrapFS = bootstrap(X(:,trainIndex),label(trainIndex),10);
23     %returned matrix bootstrapFS has one more row than X, the class label
24     %row vector
25     for j = 1:M
26
27         nnTrainInput = bootstrapFS(1:row,:,j);
28         nnTrainTarget = bootstrapFS(row+1,:,j);
29         nnTestInput = X(:,testIndex);
30         nnTestTarget = label(testIndex);
31
32         NN = newpr(nnTrainInput,nnTrainTarget,hiddenNeurons);
33         NN.trainParam.showWindow = 0;
34         NN.divideParam.testRatio = 0;
35         NN.divideParam.trainRatio = 0.7;
36         NN.divideParam.valRatio = 0.3;
37         nnComittee{j} = NN;
38         minSAE = 1000;
39         for k = 1:numIteration
40             NN = init(NN);
41             NN = train(NN,nnTrainInput,nnTrainTarget);
42             Y = sim(NN,nnTestInput);
43             SAE = sum(abs(nnTestTarget-Y));
44             if SAE < minSAE
45                 minSAE = SAE;
46                 nnComittee{j} = NN;
47             end
48         end
49     end
50     %Compute the output of committee for each fold
51     nnComitteeOutput = zeros(M,length(nnTestTarget));
52     for l = 1:M
53         nnComitteeOutput(l,:) = sim(nnComittee{l},nnTestInput);
54     end
55     %vote on the output
56     quantizedOutput = sum(round(nnComitteeOutput));
57     positive = quantizedOutput > 1;
58     negative = quantizedOutput <= 1;
59     quantizedOutput(positive) = 1;
60     quantizedOutput(negative) = 0;
61     testResult(testIndex) = quantizedOutput;
62 end
```

Summary of Publications

Thesis Title
Automatic Heart Sound Analysis for Tele Cardiac Auscultation

Author
Sumeth Yuenyong

School
School of Information, Computer and Communication Technology (ICT)

Year of Graduation
2nd Semester, Academic Year 2009

Thesis Committee:

Assoc. Prof. Waree Kongprawechnon, PhD	Chair and Advisor
Kanokvate Tungpimolrut, PhD	Committee Member and Chairperson of Examination Committee
Itthisek Nilkhamhang, PhD	Committee Member
Prof. Akinori Nishihara, PhD	Committee Member

International Conference Proceedings:

1. Sumeth Yuenyong, Waree Kongprawechnon, Kanokvate Tungpimolrut, and Akinori Nishihara, "Automatic Heart Sound Analysis for Tele-Cardiac Auscultation", *In* Proceeding of the ICROS-SICE International Joint Conference 2009 (ICCAS-SICE 2009), Fukuoka, Japan, August 18-21, 2009, 6 pages.
2. Sumeth Yuenyong, Waree Kongprawechnon, Kanokvate Tungpimolrut, and Akinori Nishihara, "A Novel Method for Heart Murmur Detection without Segmentation of Phonocardiogram", *In* Proceeding of the International Conference on Information and Communication Technology for Embedded Systems 2010 (ICICTES 2010), Pathum Thani, Thailand, January 28-30, 2010, 7 pages.

References

[1] *Thailand Health Profile 2001-2004*. Ministry of Public Health, 2004.

[2] *Fundamentals of Lung and Heart Sounds, Third Edition*. Mosby, 2004.

[3] Z. Syed, D. Leeds, D. Curthis, F. Nesta, R. A. Levin, and J. Guttag, "A framework for the analysis of acoustical cardiac signals," *IEEE Transactions on Biomedical Engineering*, vol. 1, no. 51, pp. 651–662, 2007.

[4] P. Poole-Wilson, R. A. Walsh, R. A. O'Rourke, and V. Fuster, *Hurst's The Heart*. McGRAW-HILL, 2000.

[5] C. H. Chen and P. S. P. Wang, *Handbook of Pattern Recognition and Computer Vision*. World Scientific, 2005.

[6] H. Liang, S. Lukkarinen, and I. Hartimo, "Heart sound segmentation algorithm based on heart sound envelograms," *Computer in Biology*, no. 24, pp. 105–108, 1997.

[7] ——, "A heart sound feature extraction algorithm using wavelet decomposition and reconstruction," in *Proceedings of the IEEE International Conference in Medicine and Biology*, vol. 20, 1998, pp. 1539–1542.

[8] T. Olmez and Z. Dokur, "Classification of heart sounds using artificial neural network," *Pattern Recognition Letters*, no. 24, pp. 617–629, 2003.

[9] S. Ari and G. Saha, "in search of an optimization technique for artificial neural network to classify abnormal heart sounds," *Applied soft computing*, no. 9, pp. 330–340, 2009.

[10] S. Kumar, *Neural Networks: A Classroom Approach*. McGraw Hill, 2005.

[11] D. Kumar, P. Carvalho, M. Antunes, J. Henriques, A. S. e Melo, and J. Habetha, "detection of s1 and s2 heart sounds by high frequency signatures," in *Proceedings of the 28th IEEE EMBS Annual Conference*, 2006.

[12] P. Wang, Y. Kim, L. H. Ling, and C. B. Soh, "First heart sound detection for phonocardiogram segmentation," in *Proceedings of the 27th IEEE Engineering in Medicine and Biology Annual Conference*, 2005.

[13] S. Schmidt, E. Toft, C. Graff, and J. Struijk, "Segmentation of heart sound recordings from an electronic stethoscope by a duration dependent hidden markov model," *Computers in Cardiology*, no. 35, pp. 345–359, 2008.

[14] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Networks*, no. 6, pp. 525–533, 1993.

[15] S. R. Messer, J. Agzarian, and D. Abbott, "Optimum wavelet denoising for phonocardiograms," *Microelectronics*, no. 32, pp. 931–941, 2001.

[16] C. N. Gupta, R. Palaniappan, S. Swaminathan, and S. M. Krishnan, "Neural network classification of homomorphic segmented heart sound," *Applied Soft Computing*, no. 7, pp. 286–297, 2007.

[17] S. Rajan, E. Budd, M. Stevenson, and R. Doraiswami, "Unsupervised and uncued segmentation of the fundamental heart sounds in phonocardiograms using a time-scale representation," in *Proceedings of the 28th IEEE EMBS Annual Conference*, 2006.

[18] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[19] T. R. Reed, N. E. Reed, and P. Fritzson, "Heart sound analysis for symptom detection and computer-aided diagnosis," *Simulation Modelling Practice and Theory*, no. 12, pp. 129–146, 2004.

[20] G. Amit, N. Gavriely, and N. Intrator, "Cluster analysis and classification of heart sounds," *Biomedical Signal Processing and Control*, 2008.

[21] S. Chauhana, P. Wang, C. S. Lima, and V. Anantharamanb, "A computer-aided mfcc-based hmm system for automatic auscultation," *Computers in Biology and Medicine*, 2008.

[22] R. S. MacWalter, "Heart sounds and murmurs," tayside Universtiy Hospital, Scotland http://www.dundee.ac.uk/medther/Cardiology/hsmur.html.

[23] Y. Kocabasoglu and R. Henning, "Human heart sounds," http://www.lf2.cuni.cz/Projekty/interna/heart sounds/h12/index.html.

[24] L. Glass and B. Pennycook, "Virtual stethoscope," mcGill University, http://sprojects.mmip.mcgill.ca/mvs/mvsteth.htm.

[25] R. J. Hall, "Heart sounds and murmurs," texas heart institute, USA http://www.texasheart.org/Education/CME/index.cfm.

[26] B. L., "Bagging predictors," *Machine Learning*, 1996.