Mehmet Cemal ATLIOĞLU

M.S. Thesis In Computer Engineering

June - 2012

DIAGNOSIS OF PULMONARY DISEASES FROM LUNG SOUNDS BY MACHINE LEARNING ALGORITHMS

by

Mehmet Cemal ATLIOĞLU

June 2012

DIAGNOSIS OF PULMONARY DISEASES FROM LUNG SOUNDS BY MACHINE LEARNING ALGORITHMS

by

Mehmet Cemal ATLIOĞLU

A thesis submitted to

the Graduate Institute of Sciences and Engineering

of

Fatih University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Engineering

June 2012
Istanbul, Turkey

APPROVAL PAGE

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____
Asst. Prof. Dr. Kadir TUFAN
Head of Department

This is to certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____
Asst. Prof. Dr. Nahit EMANET
Supervisor

Examining Committee Members

Asst. Prof. Dr. Nahit EMANET _____

Prof. Halil Rıdvan ÖZ _____

Dr. İhsan Haluk AKIN _____

It is approved that this thesis has been written in compliance with the formatting rules laid down by the Graduate Institute of Sciences and Engineering.

_____
Assoc. Prof. Nurullah ARSLAN
Director

June 2012

DIAGNOSIS OF PULMONARY DISEASES FROM LUNG SOUNDS BY MACHINE LEARNING ALGORITHMS

Mehmet Cemal ATLIOĞLU

M. S. Thesis - Computer Engineering
June 2012

Supervisor: Asst. Prof. Dr. Nahit EMANET

ABSTRACT

In this study, applicability of various most used feature extraction methods of the literature has been examined over lung sounds by Neural Networks and other Machine Learning Algorithms for the diagnosis of pulmonary diseases with the focus on Asthma. The sound dataset used in the study was collected by the medical experts of Gaziantep University Medical Faculty.

Mainly the study is consisted of four sub-processes such as; filtering, feature extraction, feature selection and classification. In the filtering step, the input sound dataset either used without any filtering process or various filters (digital BandPass or non-linear Teager and Median filters) were applied to the input dataset for attenuating the possible noise effects in the sound samples and eliminating the useless frequency regions, thus enhancing the classification performance of the target classification algorithms. Then in the feature extraction step, Mel frequency Cepstral coefficients (MFCC), linear predictive coefficients (LPC), time domain features and phonetic features were used to transform the input dataset into feature vectors. Feature extraction algorithms that were used for the thesis study mostly divided the sound samples into multiple “window chunks” and then produced their outputs as a series of feature sets belonging to these windows. So the unification of these multiple feature sets into a single feature vector and the determination of the feature vector length was done with the use of statistical methods such as; Mean, Standard Deviation, Skewness and Kurtosis. Then the Chi-Square feature selection algorithm was applied to feature vectors for dimensionality reduction. Lastly, the classification process was performed through a computer simulation environment and the results were plotted.

Keywords: Biomedical Applications, Pulmonary Disease Classification, Neural Networks, Lung Sounds Processing, Sound Feature Extraction, Asthma, MFCC, LPC.

GÖĞÜS HASTALIKLARININ AKCİĞER SESLERİNDEN MAKİNE ÖĞRENME ALGORİTMALARI İLE TEŞHİSİ

Mehmet Cemal ATLIOĞLU

Yüksek Lisans Tezi – Bilgisayar Mühendisliği
Haziran 2012

Tez Danışmanı: Yrd. Doç. Dr. Nahit EMANET

ÖZ

Bu çalışmada, literature bulunan birçok özellik çıkarım metodunun göğüs hastalarına ait ses-verileri üzerindeki uygulanabilirliği; yapay sinir ağları ve diğer makine öğrenme algoritmaları kullanılarak astım hastalığının teşhisi amacıyla incelenmiştir. Çalışmada kullanılan ses verileri Gaziantep Üniversitesi Tıp Fakültesi uzmanlarınca kaydedilmiştir.

Çalışma temel olarak; seslerin filtrelenmesi, özellik çıkarımı, özellik eleme ve sınıflandırma şeklinde dört alt işlem grubundan oluşmaktadır. Filtreleme kısmında, giriş sesleri hiçbir filtreleme işlemine tabi tutulmadan veya ses örneklerindeki muhtemel gürültü etkilerini azaltmak, lüzumsuz frekans bölgelerini çıkarmak ve böylece öğrenme algoritmalarının sınıflandırma performansını arttırmak amacıyla, çeşitli filtrelere (dijital BandPass veya lineer olmayan Teager ve Median filtreleri) tabi tutularak kullanılmıştır. Özellik çıkarımı kısmında, giriş seslerini özellik vektörlerine dönüştürmek için; MFCC, LPC, zaman domeni ve fonetik özellik çıkarım metotları kullanılmıştır. Çalışmada kullanılan özellik çıkarım metotları genel olarak giriş seslerini birçok küçük zaman parçacıklarına bölüp, daha sonra çıkışlarını tüm bu parçacıklara ait özellik vektörlerinin toplamından oluşan bir set olarak üretmektedirler. Bu sebeple, sözü edilen çoklu özellik vektörlerinin teke indirgenmesi ve ayrıca özellik vektörünün uzunluğunun tespiti için; Ortalama, Standart Sapma, Skewness ve Kurtosis gibi istatistiksel metotlardan faydalanılmıştır. Özellik eleme kısmında, elde edilen özellik vektörlerine Chi-Square istatistiksel eleme algoritmasının uygulanmasının etkileri incelenmiştir. Son olarak yapılan incelemeler bir benzetim ortamında gösterilip sonuçlar tablolanmıştır.

Anahtar Kelimeler: Biyomedikal Uygulamalar, Göğüs Hastalıklarının Sınıflandırılması, Yapay Sinir Ağları, Akciğer Ses İşleme, Ses Özellik Çıkarımı, Astım, MFCC, LPC.

To my parents

ACKNOWLEDGEMENT

I express my sincere appreciation to Asst. Prof. Dr. Nahit EMANET for his guidance and insight throughout the research. It was a great pleasure for me to have the chance to work with him. Thanks go to the other faculty members Prof. Halil Rıdvan ÖZ, Dr. İhsan Haluk AKIN, Prof. Onur TOKER and Prof. Bekir KARLIK for their valuable contribution to the thesis and me.

I am deeply in debt to Asst. Prof. Dr. Zoltan KATO, Asst. Prof. Dr. Arpad BESZEDES, Asst. Prof. Dr. Lazslo G. NYUL and Dr. Pluhar ANDRAS from University of Szeged for their valuable efforts during my study and research in Hungary.

I also express my special thanks and appreciation respectively to my father Uzm.Dr. Eyüp ATLIOĞLU, to my mother Esma Nuray, to my sister Neslihan and to my sweet niece Azra for their great motivation, patience and understanding.

Lastly, but in no sense the least, I am thankful to all colleagues and friends who made my stay at the university a memorable and valuable experience.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF SYSMBOLS AND ABBREVIATIONS

SYMBOL/ABBREVIATION

ADALINE	Adaptive linear element
ANN	Artificial neural network
API	Application programmer interface
AR	Auto regressive
CAFS	Centroid of audio frequency spectrum
COPD	Chronic obstructive pulmonary disease
CTFT	Continuous time Fourier transform
DAQ	Data acquisition
DC	Direct current
DCT	Discrete cosine transform
DFT	Discrete Fourier transform
DIP	Diffuse interstitial pneumonia
DOM	Document object model
DT	Decision tree
DTFT	Discrete time Fourier transform
FFT	Fast Fourier transform
FT	Fourier transform
FV	Feature vector
GMM	Gaussian mixture model
ICA	Independent component analysis
k-NN	k-Nearest neighbor
LDA	Linear discriminant analysis
LMS	Least mean squares
LPC	Linear predictive coefficient
LPCC	Linear predictive Cepstral coefficient

MAR	Multivariate auto regression
MFC	Mel frequency Cepstrum
MFCC	Mel frequency Cepstral coefficient
MSE	Mean squared error
PCA	Principal component analysis
QP	Quadratic programming
ROC	Receiver operating characteristics
SBE	Sub band energy
SD	Standard deviation
SNN	Supervised neural network
STAZCR	Short time average zero crossing rate
STE	Short time energy
STFFT	Short time Fourier transform
SVD	Singular value decomposition
SVM	Support vector machine
S.T.	Such that
WHO	World health organization
WT	Wavelet transform
XML	Extensible markup language
VQ	Vector quantization
ZCR	Zero crossing rate

CHAPTER 1

INTRODUCTION

Humanity experienced many new disorders during the last century. Modern medical science has been developing newer methods and treatment procedures each day to overcome the existing and possible future diseases (the Editors of Publications International Ltd., 2012). The pulmonary diseases are one of the major health problems of our decade. It not only affects most of the world population but it also strongly lowers the patience's life quality by blocking the inhalation mechanism. To emphasize the importance of the fact, the latest reports of the WHO (World Health Organization, 2011) can be referenced to better sense the worldwide fatal effect of pulmonary diseases over world health.

Throughout this study, by regarding all stated facts, we have worked on a solution for the diagnosis of pulmonary diseases, mainly Asthma, from the lung sounds. The sound data used in this work has been recorded by the medical experts of Gaziantep University Medical Faculty from the subject's chest region by Sony electrode condensed microphones. The digital acquisition process has been done through a two channel data acquisition (DAQ) card which was developed by Prof. Onur TOKER at Fatih University.

1.1 HISTORY OF AUSCULTATION

Centuries ago, Hippocrates was the first one to notice that breath sounds other than normal are heard over the chest of patients with pulmonary dysfunction and made a first attempt to describe them (Taplidou & Hadjileontiadis, 2010). That approach opened a huge door to future studies on the respiratory auscultation.

The study of the anatomy, physiology and pathology of the human respiratory system is known as pulmonology or respiratory medicine. One of the most important advances in the history of respiratory medicine was the development of the stethoscope in 1816 by French physician Rene-Theophile-Hyacinthe Laennec in Paris. This instrument enabled physicians to more precisely diagnose diseases of the chest and heart (Rogers K. , 2010). The first version of the stethoscope device by Laennec was made of a long wooden tube and it was monaural (Laennec, 1819). In 1841, Golding described a flexible tube stethoscope (Golding, 1840) that he used as an assistive tool where the device had one earpiece.

The invention of the binaural stethoscope is by Irish physician Arthur Leared in 1851 and the perfection of the design as a commercial product is by George P. Cammann of New York in 1852. Cammann also described the way how this new binaural product could be used for auscultation diagnosis. The refined device had one earpiece and that model has been used for more than 100 years without a major modification. In 1960, Dr. David Littmann, a Harvard Medical School professor, patented a new stethoscope that was lighter than previous models and had vastly improved acoustics (Littmann Inc. 3M, 2012).

Today mainly two kinds of pulmonary instruments are used by the physicians, namely, the acoustic and the electronic stethoscopes. The mechanism of acoustic stethoscopes is simple. A plastic covered diagram collects the sounds from its surface and transmits the sound pressure to earring nodes by the use of air filled tubes. In this method collected sound can easily be inferred by environment effects and the sound amplitude may lower due to the subject's respiratory characteristics. The electronic stethoscopes, as being the second type of pulmonary instruments, mostly rely on the idea of transmitting the sound waves into the electrical signals and then amplifying the analog signals and processing them for optimal listening. In most devices mid frequencies are amplified where low and high frequencies are attenuated.

Due the process of listening to the pulmonary chest region with a simple stethoscope device is cheap and easy in the applicability point of a physician, there are many obstacles in both devices that limit the auscultation diagnosis. First of all, the lung sounds have non-stationary characteristics which make them subjective from person to person. Moreover the procedure itself is subjective as it depends on the physician's own

hearing, experience and ability to distinguish between different sound patterns. For the acoustic stethoscopes the lower frequency band transmitted to human ear is about 120 Hz (Sovijarvi, Vanderschoot, & Earis, 2000). So even if the physician is careful and talented enough, still there can be inaudible regions in the target signal causing the physician to miss many important sound patterns.

To overcome the problems in the diagnosis process of pulmonary inhalation sounds, many physicians strongly rely on some other tests (Rogers K. , 2010) which can be listed as; pulmonary function test (PFT) (which is also known as spirogram test), chest X–ray, arterial gas analyses, lung ventilation/perfusion scan, bronchoscopy, mediastinoscopy and mediastinotomy, thoracentesis, thoracotomy, thoracoscopy and video-assisted thoracoscopic surgery, challenge test, pleural biopsy, transthoracic needle biopsy, tube thoracostomy and allergy testing (National Asthma Council Australia, 2011), (American Thoracic Society, 2012), (The Lung Association of Canada, 2012), (The Merck Manual, 2012).

The evolution of the pulmonary acquisition instruments is not complete. From wooden tubes to modern electronic designs the final purpose of deriving higher quality sound for catching the difficult patterns of lung sound biometrics will always be on the target of modern pulmonary studies.

1.2 PURPOSE OF STUDY

The main purpose of this study is establishing an efficient algorithm for the early intelligence and diagnosis of the pulmonary diseases, mainly asthma, from the lung sounds. For the desired algorithm to work efficiently in real life conditions, the input sound data set including environment noise has been freely used without regarding any noise removal concerns. Since the lung sound acquisition procedure through a microphone cannot be performed in a professional studio environment with the expectation of exact silence, the best methodology should also be able to isolate the true disorder patterns from the environment noise without the need of any special needs as the medical expert’s ear does.

As the second main purpose of this study, examining the efficiency and the applicability of the most used feature extraction algorithms over medical data is aimed. If the sound data is regarded as a vector of thousands of numeric values in computer environment, the mapping of this numeric vector into a fixed size, smaller form by a simple extraction algorithm would be a time saving and efficient progress. After transforming the sound data into desired smaller form, deriving the highest classification accuracy for the neural networks, linear classifiers and other machine learning algorithms has lastly been aimed.

1.3 STUDY AREA

The thesis study is in touch with many multidisciplinary expertise fields such as internal medicine, sound processing, machine learning, artificial intelligence, pattern recognition, neuro-cognitive sciences, biomedical engineering and statistics. Because the border of the study area is that wide, strong background knowledge is needed for such a study to produce applicable results. For simplifying the overall process against the huge theoretical needs, an analytic progress through computer based simulation environment and an experimental reasoning methodology has been used in each step of the study.

As a general summary, the target is improving human life quality with the use of computers and machine learning tools as expert systems. But from an upper point of view the target is to resolve the problems by first analyzing the answers of the medical experts, for the purpose of catching the hidden patterns, and then using these patterns for producing a solution to future harder problems where experts may fail or struggle.

1.4 ORGANIZATION OF THE THESIS

The thesis is consisted of eight individual chapters. The organization structure of the thesis can be broken down as follows; Chapter-1 gives brief information about the study and describes the main purpose of the work. Chapter-2 provides a detailed review of the existing techniques and studies in the literature. Chapter-3 gives the theoretical

information for the machine learning algorithms used in the study. Chapter-4 explains the fundamentals of signal processing techniques. Chapter-5 overviews both time domain and frequency domain sound feature extraction methods that were used through experiments. Chapter-6 underlines the basic statistical methods which were used in the thesis study. Chapter-7 provides the details of the building blocks of the experiment environment, describes the data set preprocessing steps, gives the numerical and visual results of the system along with implementation details and also presents a comparison between the optimized and raw input vector usage. Finally, Chapter-8 provides the conclusion and makes suggestions for the future work.

CHAPTER 2

LITERATURE REVIEW

Diagnosis of pulmonary dysfunctions has a long history in the literature. Many researchers worked on the subject with different methodologies. Since any kind of analog transducer outputs can be represented in the computer environment as binary streams, the work on the sound streams mostly depended on the previous linear and nonlinear classification, sound processing and pattern recognition algorithms. In this chapter detailed information of literature on the most recent methods will be presented.

In 1984, Cohen et al. (Cohen & Landsberg, 1984) proposed their study on the analysis and automatic classification of breath sounds. According to the study, the main goals are twofold: to characterize the various breath sounds quantitatively and to provide an automatic classification technique for the various types of sounds with the goal of providing the physician a strong diagnostic support device. In the study, the linear prediction (LPC) and the peak factor coefficients (PFC) were used as feature extraction algorithms. For the classification phase, a twofold classification schema is used due to weak classification results of single classification experiments. Lastly the design of a future embedded device is also argued with the actual study on the subject.

In 1992, Shabtai-Musih et al. (Shabtai-Musih, Grotberg, & Gavriely, 1992) proposed a recognition algorithm for the detection of the “wheeze patterns” in the respiratory sounds. In a general view this algorithm aims to detect the presence of wheeze patterns in the sound, rather than pointing to any pulmonary disease. The first step of the algorithm is taking 128 samples short-time Fourier transform (ST-FFT) of the signal with an overlapping ratio of 108 points (which is about 85%). Then the mean is subtracted from all points in the spectrum and thus the spectrum is normalized to its standard deviation. Next the presence of peaks in the spectrum is analyzed so that the

peaks which are greater than “3.5 standard deviations (SD)” are evaluated according to the criterion that is designed to distinguish instantaneous (narrow) peaks from wide peaks, steps, ramps and multiple close peaks. The score is incremented if the evaluated peak meets the criterion. A “score threshold” is taken as 3 (of maximum7), and the scores greater than 3 are accepted as wheezes. Lastly the frequencies of accepted wheezes are recorded in their corresponding temporal position. For a better progress, the “Improved Shabtai-Musih Algorithm” scans the signal using time segments of 256 samples with an overlap ratio of 50%. To eliminate the spectral leakages, Hanning windowing technique is used. The mean value of each 250 Hz frequency band is obtained and subtracted from this frequency segment in order to compensate for spectral irregularities of respiratory tracheal sound transmission. Normalization using the standard deviation is also performed.

In 2006, Meng-Lun et al. (Meng-Lun, Jen-Chien, & Feng-Chia , 2006) worked on a respiratory data set of 16 subjects. They divided the data set into two groups. The first group consisted of 12 nonsmoking asthmatic patients who all had various kinds of wheeze episodes. The second group was composed of 4 subjects without any reported respiratory pathology. The sounds were acquitted from the thorax region of the subjects in sitting position with a period of 7 to 10 respiratory cycle’s per-person. Their acquisition hardware consisted of an Omni directional electrical condenser microphone coupled with a low power pre-amplifier. The system then followed by a band-pass anti-aliasing filter consisting of a high-pass Bessel (roll of > 18 dB) and a low-pass Butterworth filter (roll of > 24 dB). Lastly the digitalization was done by a commercial sound card with a standardized sampling rate of 44.1 kHz. In the digital signal processing part, rather than rule based wheeze detection algorithms, they considered the spectrogram as an image file and filtered out the [100-1600 Hz] frequencies in the spectrogram as the out of interest regions. Then set a limiter to mean spectral energies and applied an image processing technique (a bilateral filter to smooth the image data point) to object wheeze directly. Their work proposed a “Yes or No like Decision Support System” to the doctors for simplifying the detection of wheeze patterns in the sound. The algorithm was tested on more samples in the NTU hospital with a recorded sensitivity and specificity above 89%.

Next in 2006, Martinez-Hernandez et al. (Martinez-Hernandez, Aljama-Corrales, Gonzalez-Camarena, Charleston-Villalobos, & Chi-Lem, 2006) proposed their study for multichannel acquisition of lung sounds by a microphone array. In their work they have attached a 5x5 sensor group, which was vertically and horizontally separated by 5 to 7 cm, to the back of the subjects. They have acquitted the sounds by a digital card with a sampling frequency of 10 KHz for 15 sec.'s while the subjects were seated. A digital filter applied to the [75-2000 Hz] frequency bands for reducing the interference of heart and muscle sounds. Then each inspiratory phase were extracted from the sound and divided into 30 equal windows with an overlap ratio of 25% between them. The lung sounds feature extraction process was done by a multivariate auto regressive (MAR) model and dimensionality reduction of the feature vectors (FV) was also done by SVD and principal component analyses (PCA). Lastly a supervised neural network (SNN) was used for the classification of the derived feature vectors. The algorithm aimed to classify diffuse interstitial pneumonia (DIP) disease, which can be diagnosed by the existence of crackles from the normal breath respiratory sound. As a result the proposed algorithm achieved an accuracy value of 82% for the diagnosis of DIP disease from the pulmonary sounds.

Later in their 2007 work, Feng and Satttar have presented a new method to identify wheezing and normal breathing from tracheal sounds (Feng & Satttar, 2007). In the work instead of standard frequency-domain features, authors introduced a "more straightforward time-domain features set" generated by their adaptive scheme. Their method first performed a time-frequency analysis over the data set and used "recursively measured averaged normalized instantaneous kurtosis" for identification of the wheeze and normal breath sounds. Also in their proposed scheme the sample entropy, which measures the predictability of the current amplitude values of a physiological signal based on its previous amplitude values, was made use of as a statistical parameter. The sharp drops during the normal breath sounds and the superimposed shape of histogram for wheeze signals provided the authors a clearer view about the superimposed nature of wheeze on normal breath. This statistical characteristic was kept for their further researches to separate adventitious sounds from normal breath.

In 2009, Sergul at al. (Aydore, Sen, Kahya, & Mihcak, 2009) proposed a new method in their study. The aim of this study is the classification of wheeze and non-wheeze epochs within the respiratory sound signals that are acquired from patients with asthma and COPD. The features proposed in this study are kurtosis, Renyi entropy and mean-crossing irregularity calculated in the time domain and f50/ f90 ratio calculated in the frequency domain. Fisher discriminant analysis and Neyman Pearson hypothesis testing were applied for classification and detection. After the error rates have been calculated for the training set, the test error has been calculated for the same data set using leave-one-out method. The sound data used in the work was collected from 4 male and 3 female subjects with an age range from 33 to 67. By visually inspecting the time expanded sound signals, an expert labeled the wheeze portions and non-wheeze portions that are at 15 sec lengths. A total of 246 wheeze and non-wheeze portions were thereby labeled and used in the study. Then for each 246 portions, 4 features were extracted. After the extraction of four features for each window from both classes, the Fisher discriminant analysis was applied in order to separate the two classes in one-dimension. The algorithm achieved a correctness of 93.5% as a test result. And as an extension to present linear classifiers, the future use of non-linear classifiers, the use of a larger data repository and the addition of more feature sets to the experiment has also been discussed.

In 2010, Sen at al. (Sen, Saraclar, & Kahya, 2010) proposed a study to devise a methodology to estimate and depict the source locations of “respiratory adventitious sound components” (particularly crackles) in the lungs, because of their association with certain pulmonary diseases. The sound data used in the work belonged to a subject who was diagnosed with cystic bronchiectasis in the lower left lung. The data was recorded using a 14-channel respiratory sound acquisition (DAQ) device. The system composed of 14 electrode microphones attached on the posterior chest wall (seven microphones per side), plus an analog amplifier-filter unit to process microphone output signals (an instrumentation amplifier with a gain of 100 and an 80−4000 Hz band pass filter), a DAQ card (NI-6024E) to transfer the processed signals to the computer after digitization, and a notebook computer executing an interface developed in LabVIEW environment. The sampling rate was chosen 9600 Hz with duration of 15 seconds per single DAQ session. After the acquisition completed, the multi-channel data segmented into pieces of 1000 samples. Then for all channel segments, a prior test was applied to

decide automatically whether there exists at least one channel that is characterized by the existence of crackles. Only segments with some evidence of the existence were inputted to the source separation algorithm (namely, basic independent component analysis (basic ICA)). Outputs of ICA were in turn tested to decide which one of them is the true crackle source signal by an evaluation of the mixing coefficients in a center of weights approach. Lastly by the use of a Bayesian classifier, the algorithm performed an accuracy value of 85%.

In 2011, Shuiping at al. (Shuiping, Zhenming, & Shiqiang, 2011) proposed their algorithm for the design and implementation of an audio classification system with support vector machines (SVM). The main aim of this study is seperating the music based audio clips from the data based voice clips with the use of both time and frequency features within a given data base. In the experiment, the original audio repository included 2500 clips, where the 1200 clips are voice data and the resting 1300 clips are music clips. 800 voice clips and 800 music clips were chosen for the training set. The rest 300 voice clips and 400 music clips were used as the test set. First the audio clips were segmented into smaller window chunks in the data pre-processing step. Then the authors used Short-Time Average Zero-Crossing Rate (STAZCR) and Short-Time Energy (STE) algorithms for deriving the time domain features. Next, the centroid of audio frequency spectrum (CAFS), Sub-Band Energy (SBE) and " both the mean and variance" of mel frequency cepstral coefficents (MFCC) were used as frequency domain characteristic parameters. By forming the extracted features together, a SVM based audio classification system has been designed. Results of the study shows that a classification accuracy of 92% has been achieved with the choosen feature sets. This work has an importance of combining both the time domain and frequency domain features into the same fature vector, and the experiment shows that the results are quite promising.

In 2012, Ai et al. (Ai, Hariharan, Yaacob, & Chee, 2012) proposed a study for the classification of speech dysfluencies with the use of MFCC and LPCC features. For constructing the repository 39 speech samples were taken from the University College London Archive of Stuttered Speech (UCLASS). Content of the speech samples are ''One more week to Easter'' and ''Arthur the rat'' (90% of these are mono-syllable words). The two types of dysfluencies in stuttered speech such as repetitions and

prolongations were identified and segmented manually by listening to the recorded speech signal. The segmented speech samples were down sampled from 44.1 kHz to 16 kHz for speech pre-processing purpose because most of the salient speech features are within 8 kHz bandwidth. The sampled speech signals were pre-emphasized with a simple first order high pass filter for the purpose of evening the spectral energy envelope by amplifying the importance of high-frequency components and removing the DC component in the signal. The pre-emphasized signal was divided into short frame segment using Hamming window. Later two speech parameterization techniques were applied to each window, namely, MFCC and LPCC to derive the frequency domain feature vectors. In the study, 25 MFCCs and 21 LPCCs were extracted to discriminate the two types of dysfluencies. For further analysis, three parameters were varied such as; frame length changed from 10 to 50 msec. with adjusting overlapping window between no overlap (0.0%), 33.33%, 50% and 75% and a value of the first-order pre-emphasize is varied from 0.91 to 0.99. For the classification purpose two classifiers were used, namely, k-Nearest Neighbor (kNN) and Linear Discriminant Analysis (LDA). Conventional validation was performed with 70% of data and 30% of data used for testing. Data in both testing and training sets were normalized in the range [0, 1] and shuffled randomly. As main result, derivation of the feature vector by LPCC slightly outperformed MFCC in all situations. The best configuration of 21 LPCC features has shown the best accuracy of 94.51% where the optimal configuration of 25 MFCC features has presented the best accuracy of 92.55%. This work has such an importance that, the replacement of feature extraction algorithm MFCC by LPCC improves the output accuracy of the classification process both for the kNN and LDA classifiers.

From the richness point of used algorithms, a more comprehensive study was proposed by Prof. Dr. Mohammed Bahoura (Bahoura, 2009) for the respiratory sounds classification into normal and wheeze classes. In this study, the feature extraction methods and the modeling techniques which have been proposed in last two decades to classify respiratory sounds were presented and various combinations of these methods were compared using receiver operating characteristic (ROC) curves. The feature extraction methods based on, Fourier transform (FT), linear predictive coding (LPC) - which is also called auto-regressive (AR) modeling, wavelet transform (WT) and mel frequency Cepstral coefficients (MFCC) in combination with the classification methods

based on, vector quantization (VQ), Gaussian mixture model (GMM) and artificial neural network (ANN) were evaluated. Also the use of an optimized threshold to discriminate the wheezing class from the normal one was proposed. Finally, a post-processing filter was suggested to considerably improve the classification accuracy. The sound database was constructed from real respiratory sounds obtained from various sources such as; sounds that recorded on healthy and asthmatic patients, the R.A.L.E database CD, the ASTRA database CD, and various internet sites of laboratories working in the field. To ensure a digitizing standard all sounds were down sampled to 6000 Hz and normalized in amplitude. The normal class was obtained from sounds recorded on 12 healthy subjects, while the wheezing class was obtained from sounds recorded on 12 asthmatic subjects. The wheezing sounds were manually pre-processed in order to eliminate eventual non-wheezing segments with Adobe Audition software by listening and visualizing their spectrograms. Then the sounds were uniformly divided into 50% overlapping segments consisting of 1024 samples each and Hamming window was applied to all segments respectively. The training and testing phases were performed on these respiratory sound segments. As a main point of the study, the presence or absence of wheezes in the tested segment was queried, not their number nor their frequencies. In the training phase, sounds recorded on 11 subjects of each class were used while the sound recorded on the remaining one was used in the recognition phase. This process was repeated for each sound with leave-one out method. The obtained result has shown that, the proposed approach B-MFCC with GMM outperformed all other commonly used methods with a correctness of 91.9%. A significant performance improvement was obtained by smoothing the score function by reducing false detections. Also the author stated that the proposed system could be used to quantify the severity of airway obstruction by computing the proportion of the respiratory cycle occupied by wheezing.

CHAPTER 3

MACHINE LEARNING ALGORITHMS

The field of machine learning concerns the computer programs as well the design and realization of concrete algorithms that can imitate the learning behavior of human. The main aim for the scientists is to make the computers think like human beings. For a more formal definition, machine learning is a branch of artificial intelligence which includes many related disciplines such as pattern recognition, data mining and neural networks. As witnessed in recent years, chess playing computers, target tracking missiles, walking robots, self-flying planes and automatic answering machines can be listed as some products of the studies in the field.

The most fundamental concept of machine learning is the process of learning. The studies on the human and animal learning structure are the main guidelines for the machine learning algorithms. With a closer view, learning (Alpaydin, 2009, pp. 4-43) is an interaction between the learning agent and the environment with or without the help of a teacher. Basically the learning process can be categorized as; supervised learning (teacher assisted), unsupervised learning (non-assisted learning), reinforcement learning (feedback from rewards) and transduction.

For a better sense, some examples can be given for each learning paradigm. In supervised learning, the agent is first trained by a training set where this process is called as the training phase. And in the test phase the agent is expected to answer the questions that were never seen before. Classification algorithms are the specific types of supervised learning paradigm. In unsupervised learning the agent is not trained with a training set but expected to answer the questions by its own findings. Clustering algorithms can be given as the examples of this type of learning where some similarity measures are used by the agent to distinguish between the query points.

In reinforcement learning the agent first acts and then receives a feedback from the environment and then by the help of this interaction it discovers the answer. This paradigm can be regarded as learning from the experiments by positive rewards. Lastly in the transduction, the agent is first trained by a training set and then expected to answer "just some questions" in the test set. At that point making a generalization is not the desired aim where the induction, the reverse of transduction, oppositely aims making generalizations from a set of observations.

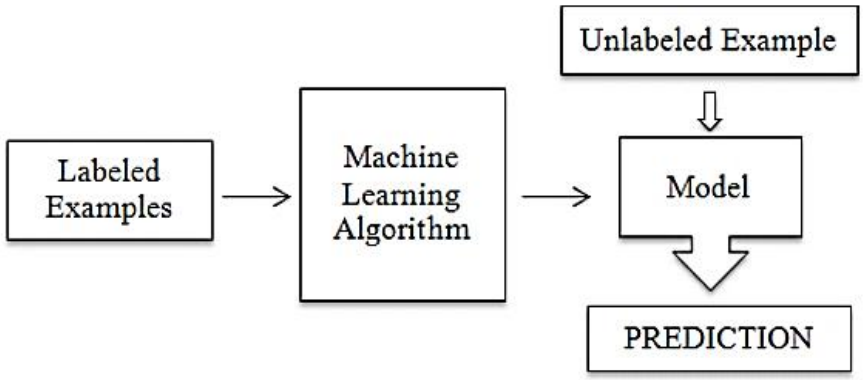


Figure 3.1 Basic elements of a supervised learning process.

There are also models of learning which consider more complex interactions between a learner and the environment. The simplest case is when the learner is allowed to query the environment about the output associated with a particular input. The study of how this affects the learner's ability of learning different tasks is known as the query learning. In learning models another type of variation is the way in which the training data are generated and how they are presented to the learner. For example, there is a distinction made between the *batch learning* (in which all the data are given to the learner at the start of learning), and the *on-line learning* (in which the learner receives one example at a time and gives its estimate of the output before receiving the correct value). In on-line learning the agent updates its current hypothesis in response to each new example and the quality of learning is assessed by the total number of mistakes made during learning (Cristianini & Taylor, 2000).

3.1 LINEAR CLASSIFIERS

A learning machine using a hypothesis that forms the linear combinations of the input variables is known as the linear learning machine. A linear learning machine produces an output model by simply evaluating the linear combinations of the input variables. This process can be regarded as drawing a straight and non-flexible line within the observed data. Basically an algorithm that maps all its input points to a linearly separated dimension is called a linear function.

$$sum = \sum_{i=1}^{n} w_i x_i + b \tag{3.1}$$

$$f(sum) = \begin{cases} -1, & sum < 0 \\ +1, & sum \geq 0 \end{cases} \tag{3.2}$$

A linear function *f(x,)* is frequently used for binary classification such that: if the *sum* (3.1) is greater than *zero*, then assign the multidimensional vector *x* to the positive (+) class, else assign it to the negative (-) class (3.2). The basic geometric interpretation of the linear classification is as follows:

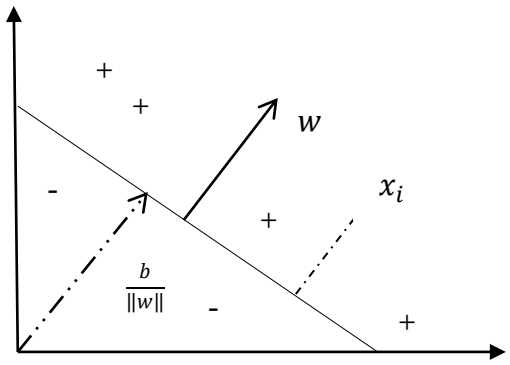


Figure 3.2 Linear classification - geometric interpretation

3.1.1 Perceptron

The problem for the linear learning machines is that, the agent needs a linear function to discriminate between the output classes to perform the reasoning. To fit a linear function then the problem turns into finding some new coefficients, namely, weight vector (*w)* and the adjusting bias *(b)*. To solve the problem of manual adjustment of the parameters, a self-adjusting solution should be found. In the machine learning literature, one of the older approaches to the problem is called the perceptron algorithm which was proposed by Frank Rosenblatt, an American psychologist, in 1956.

Mainly the perceptron was first proposed (Rosenblatt, 1962) as a model to the nerve cells in retina. In the original Rosenblatt model the computing units are threshold elements and the connectivity is determined stochastically. Learning takes place by adapting the weights of the network with a numerical algorithm. Rosenblatt's model was refined and perfected in the 1960's and its computational properties were further analyzed in details by Minsky and Papert (Minsky & Papert, 1969).

$$\omega_1 x_1 + \omega_2 x_2 + \theta = 0 \tag{3.3}$$

$$x_2 = -\frac{\omega_1}{\omega_2} x_1 - \frac{\theta}{\omega_2} \tag{3.4}$$

The single layer network, namely perceptron, represents a linear discriminant function. Given by the equation (3.3), the separation between two target classes is a straight line. The weights determine the slope of the line and the bias determines how far the line is from the origin (3.4). As a disadvantage, the perceptron doesn't care the optimal separability while drawing the separation line between the target classes, leading the separation line to be closer to one of the class samples rather than keeping the border equal to each side.

3.1.1.1 Perceptron Learning Algorithm

As being learning machines, perceptron's can learn as well. As a fundamental part of the structure, the Data is kept within the weights of the perceptron. For each given new input vector; $\vec{x} = \{x_1, x_2, \dots \dots x_n\}$ the inner weights; $\vec{w} = \{w_1, w_2, \dots \dots w_n\}$ are updated upon a learning algorithm. In that manner the learning occurs with the correction of the weights with the use of a given training set. This learning type is clearly supervised learning.

$$\omega_{i+1} \leftarrow \omega_i + \Delta\omega \tag{3.5}$$

$$\theta_{i+1} \leftarrow \theta_i + \Delta\theta \tag{3.6}$$

Learning algorithm for the perceptron is called the 'perceptron learning rule' which is an iterative procedure that adjusts the weights. First a learning sample is presented to the network and then depending on the produced error the new weight value is computed for each weight by adding a correction amount to the old value (3.5). The threshold is updated (3.6) in a same way (Kröse & Smagt, 1996) as follows:

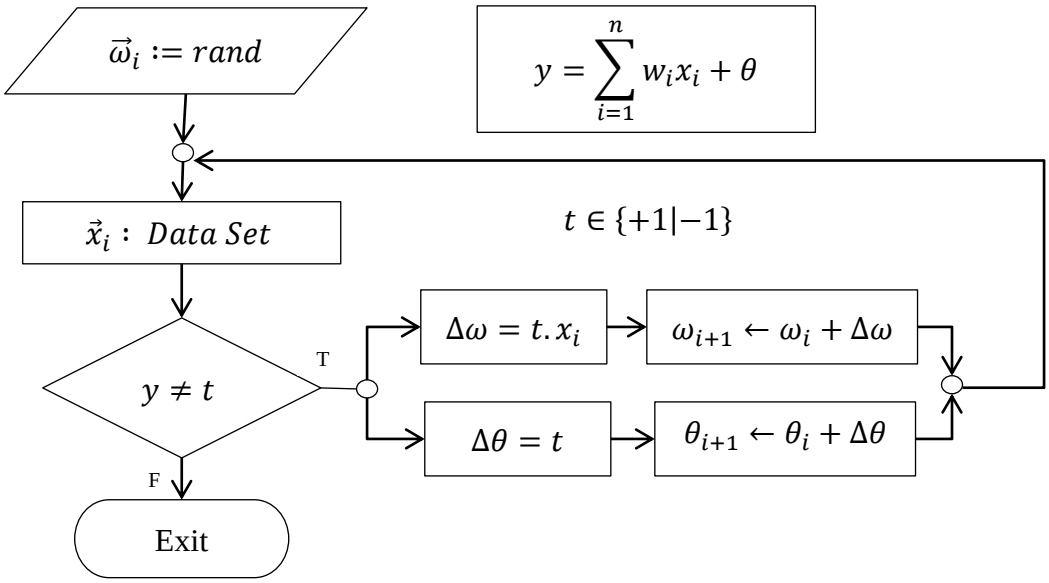


Figure 3.3 Perceptron learning algorithm.

3.1.2 Adaline

An important extension to the Rosenblatt's classical perceptron is "Adaptive Linear Element" ADALINE (Widrow & Hoff, 1960) which was proposed by Prof. Widrow and his graduate student Ted Hoff at Stanford University in 1960. It is inherently based on the McCulloch–Pitts logical threshold unit and the perceptron. The Adaline is consisted of a weight, a bias and a summation function. The main difference between Adaline and the Rosenblatt's classical perceptron is that, in the learning phase the weights are adjusted according to the continuous raw outputs. In the classical perceptron, the output is first passed to the activation function (signum) and the function's output is used for adjusting the weights. Thus it can clearly be said that the Perceptron has binary output, where the Adaline algorithm has continuous valued output during the training process. The visual representation of the Adaline (Widrow & Hoff, 1960, p. 102) is as follows:

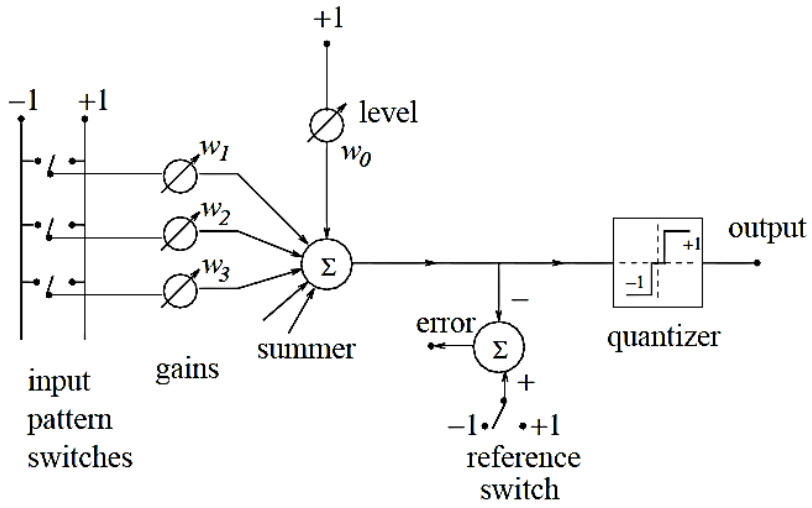


Figure 3.4 The Adaline - circuit representation.

The purpose of the Adaline device is to map a given value *y* to a desired class, consisting of +1 or -1, at its output when a set of vector values are applied to its inputs. The problem is the determination of weights in such a way that the input-output response is correct for a large number of arbitrarily chosen signal sets. If an exact mapping is not possible, at least the closest result should be derived which minimizes the average error.

3.1.2.1 Delta Rule (LMS)

For the Adaline, Widrow introduced the delta rule which is a self-adaptive learning process used for adjusting the weights during the training phase.

$$y^p = \sum_{i=0}^{n} x_i . \omega_i \qquad \textit{Actual output for pattern p} \qquad (3.7)$$

$$\varepsilon = d^p - y^p \qquad \textit{Error} \qquad (3.8)$$

$$\varepsilon = d^p - y^p = 0 \qquad \textit{Stop criteria} \qquad (3.9)$$

$$\omega_{i+1} \leftarrow \omega_i + \Delta\omega \qquad \textit{Weight update rule} \qquad (3.10)$$

$$\Delta\omega = \eta . \varepsilon . x_i \qquad \textit{Weight correction value} \qquad (3.11)$$

According to the least mean squares (LMS) algorithm, with another saying delta rule, first some random weights are assigned to the system. Then for a given input vector $(\vec{x})$, the actual output is computed according to (3.7). Next, the error which is the difference between the desired output and actual output is computed (3.8). If the stop criterion is not met then the weight vector $(\vec{\omega})$ is updated according to (3.10) with respect to the weight correction values stated in (3.11).

Training continues until the stop criterion (3.9) is met for each pattern. In practical use the error may not be absolute zero. For that reason stop criterion is checked by the comparisons of the squared error with a given closeness measurement. LMS rule assumes the squared error to be as small as possible.

While computing the weight correction value as a second important point, a percentage of the error is taken into consider for each training step. This ratio is simply named as Eta (η), which is also known as the "step size" in control theory and the "learning rate constant" in machine learning community. Since the eta is between 0 and 1, the adjustment of eta is an optimization problem that directly affects the number of training cycles and the convergence of the learner.

3.2 NEURAL NETWORKS

Neural networks are supervised classifiers that are mostly used for nonlinear classification. They inherit the working dynamics of biological nerve cells which are called neurons. In electrical and computer engineering for emphasizing the distinction between the biological nerve cells and the computational neurons, these networks are called artificial neural networks (Haykin, 1998). Since the mechanism of the brain is not fully understood, ANN's are just some abstractions to biological studies.

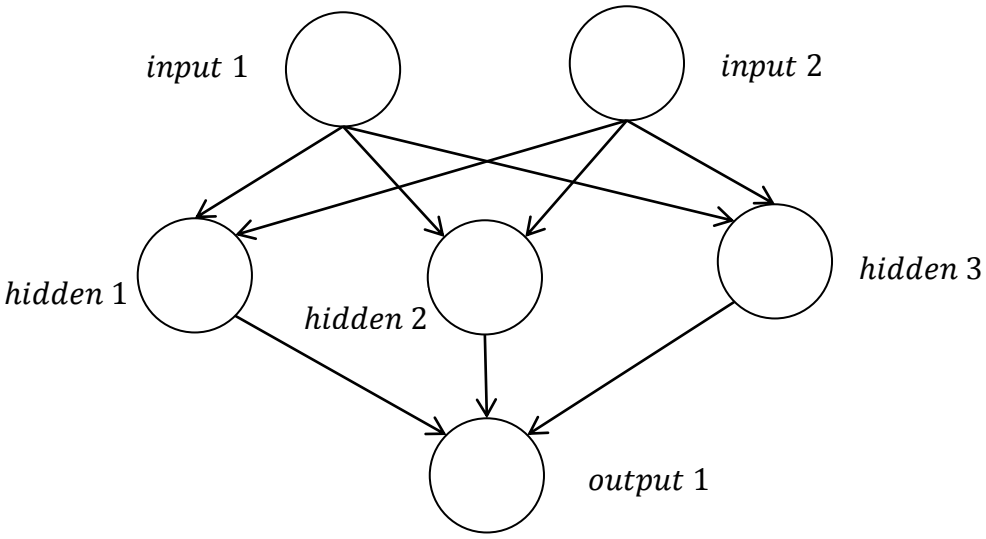


Figure 3.5 A neural network with a single hidden layer.

Since the structure is called a network, the neurons are organized within a specific wired architecture. The system accepts a feature vector at its input as input layer and the classification results are produced at the nodes of the output layer. Up to the design, there can be hidden layers between the input layer and the output layer for enhancing and optimizing the classification results. Starting from the input layer, within a directed graph, each node is connected to succeeding layer's individual nodes with a single line which represents the weights, where in the biological nerve model the location of the information is in the strength of the connection between the dendrites and axons, namely synapses. There are no connections within a layer. This one-to-all connection paradigm continues until the last hidden layer. From each last hidden layer node to each output layer node, the connection is done in a same "one-to-all" wiring paradigm.

For ANN's, the number of hidden layers and the number of nodes for each hidden layer is totally abstract. Actually any artificial neural network is valid as long as it has an input layer and at least one node in the output layer. There is no golden rule stating any concrete design criteria for the architecture. Choosing optimal number of hidden nodes per hidden layer is one of the most difficult problems when using NN's. Upon the practitioner's observations on the classification accuracies of the training set, the number of hidden layers and corresponding internal node amount is determined experimentally. Using too few hidden nodes may result to under fitting, pointing to the fact that the learning process of the target function couldn't be achieved. Or the best result is achieved without using too many hidden nodes. Sometimes as a result of using too many nodes the over fitting may occur, meaning that the network has just memorized the training set rather than learning the patterns in it. What is clear for the network is the number of output nodes. Since the output nodes determine the classification result, exactly one output node is needed for each class label.

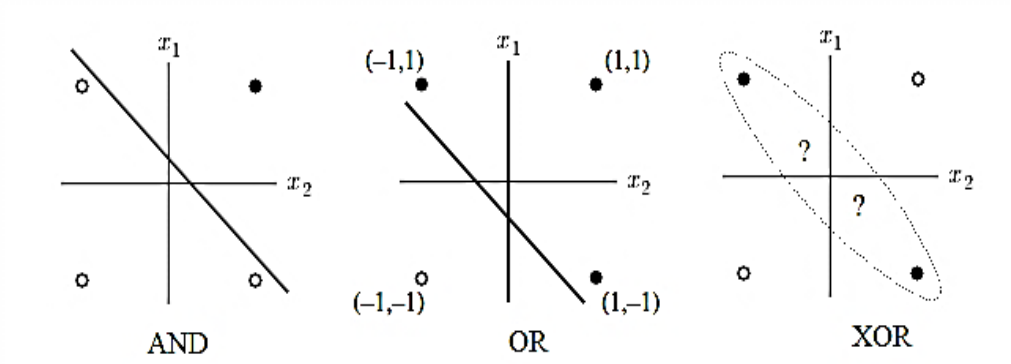


Figure 3.6 Geometric representation of linear separability problem.

Any question for the need of a hidden layer is answered by Minsky and Papert in their 1969 book *Perceptrons* (Minsky & Papert, 1969). As a simple sum up, the problem with the classical perceptron is the linear separability problem (Kröse & Smagt, 1996, p. 29), which is also known as XOR problem. A single perceptron can classify the outputs of primitive Boolean functions since they are linearly separable. For the XOR problem at least two perceptrons are needed since the separation of the input space needs two distinct lines. That can be achieved by using a hidden layer consisting of two nodes in between the input and output layers (Mitchel, 1997).

3.2.1 Activation Functions

The main aim of an activation function is scaling the output into a proper range. Each neural network node uses an activation function to produce their scaled output from the computed raw outputs with the use of a nonlinear function. As a part of the architecture the selection criterion of the activation function is abstract. Many different types of non-linear continuous functions can be used for the architecture (Duch & Jankowski, 1999). In basic terms, differentiability is the main requirement for an activation function to satisfy. The sigmoid (3.12) and hyperbolic tangent (3.13) functions can be given as examples to continuously differentiable and nonlinear activation functions which are mostly used in multilayer perceptrons.

$$F_c(x) = \frac{1}{1 + e^{-cx}} \qquad where, \qquad F_c : \mathcal{R} \to (0, 1) \tag{3.12}$$

$$f(x) = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3.13}$$

For the sigmoid function the constant parameter c can be selected arbitrarily and its reverse, c^{-1} is called the temperature value in stochastic neural networks. The value of c determines the shape, so called the sharpness of the sigmoid function. Higher values of c brings the shape of the sigmoid closer to step function. In theory c can take values between the ranges $-\infty$ to $+\infty$. For the limit $c \to \infty$ the sigmoid converges to a pure step function at the origin and for the limit $c \to -\infty$ the sigmoid converges to a straight line which can be denoted by $y = \frac{1}{2}$. To simplify the overall computations taking $c = 1$ is possible for the sigmoid activation functions.

One important point for the sigmoid functions is that they only return positive outputs. If the neural network is desired to return negative values then sigmoid function would be unsuitable. For that reason it can be replaced by hyperbolic tangent functions. Simply hyperbolic tangent function is the replacement of the sigmoid function into a region of the graph where it can return values less than zero.

3.2.2 Back Propagation Algorithm

Back propagation algorithm is the learning process of the artificial neural networks. Simply the main idea behind the process is first comparing the output of the system with a given target to derive the error and then distributing the output layer's error gradually back to the hindmost hidden layer. Since the network collectively produces the error which is experienced at the output, the algorithm considers each node's contribution on this unwanted result. From the point of application, the back propagation algorithm can be regarded as a generalized version of the delta rule (the LMS algorithm) and it can be used with any feedforward network having a differentiable activation function.

While training the neural network there are two basic steps which can be expressed according to the direction of the data flow. The data flow in a neural network is bidirectional. The first direction is from input layer to output layer which is in feedforward manner. We can name this process as the *forward-pass*. The second direction is from output nodes to the first hidden layer which is in a feedback manner. Similarly we can name this process as the *backward-pass*. Each forward-pass is followed by a backward-pass during the training procedure.

In the forward-pass, to train the neural network, first the weights in the network are assigned some small random values and then the network is fed with samples from the training set. In a cross distribution manner each feature of the input vector is represented to all nodes of the subsequent layer. The subsequent layer nodes receiving the stimulus first multiply their weights with the given feature values and then sum them together to form the raw output. Then this raw sum is passed to the inner activation function to produce the scaled final output. The procedure is repeated until the output layer where the classification occurs. For each output node's stimulus, there is a numeric target denoting the target class value. At that point the *error* can simply be defined as the difference between the desired value and the actual output.

As a standard, the back propagation algorithm uses mean squared error (3.14) to evaluate its output. With the determination of the network's error, the first pass is completed if the found error meets the criterion of a predefined range. Assuming that

the activated output from $node\ i$ is y_i but should be d_i, then the mean squared error (MSE) can be defined as:

$$(MSE) = \frac{1}{2}(y_i - d_i)^2 \tag{3.14}$$

The back propagation algorithm aims to keep the output error as small as possible. To ensure this aim, in the backward-pass, the error is propagated back from the foremost layer nodes to hindmost hidden layer nodes. Each node receiving its individual error, updates its weights with the basic weight update rule (3.15) defined by the back propagation algorithm.

$$\Delta\omega_{ji} = -\eta \frac{\partial E}{\partial \omega_{ji}} \tag{3.15}$$

Considering a simple neural network, where the *node i* is an output node, *node j* is a hidden layer node and the *node k* is an input node; ω_{ji} is the weight between $node\ i\ and\ node\ j$ and ω_{kj} is the weight between $node\ k\ and\ node\ j$. Similarly the node outputs are $y_i, y_j\ and\ y_k$. The activation functions are $S_i, S_j\ and\ S_k$ and the desired outputs are also $d_i, d_j\ and\ d_k$.

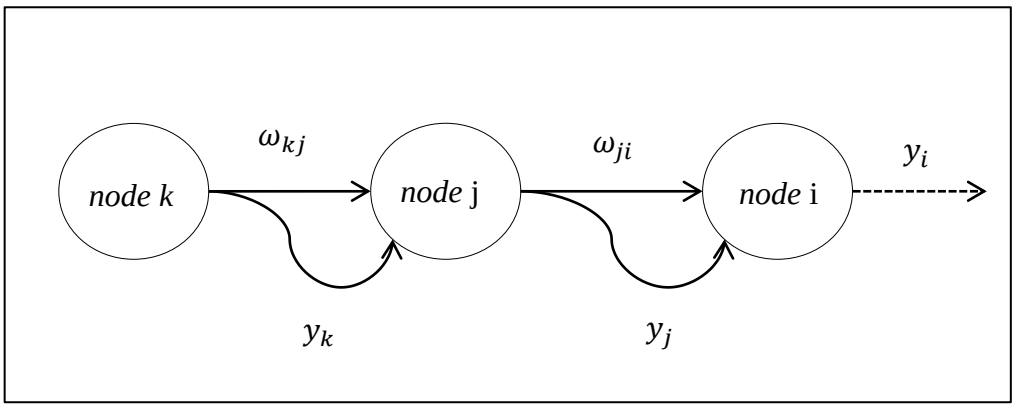


Figure 3.7 A simple neural network branch.

To adjust the weights back propagation algorithm assumes two cases for determining the weight update equations. Upon this assumption a node can be an output or a hidden layer node. In the first case, for an output node i, the weight update rule can be expressed with (3.20), if the (3.17), (3.18) and (3.19) are put in the equation (3.16).

$$\Delta\omega_{ji} = -\eta \frac{\partial E}{\partial \omega_{ji}} = -\eta \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial S_i} \frac{\partial S_i}{\partial \omega_{ji}} \tag{3.16}$$

$$\frac{\partial E}{\partial y_i} = \frac{\partial}{\partial y_i}\left(\frac{1}{2}\sum_m (d_m - y_m)^2\right) = -(d_i - y_i) \tag{3.17}$$

$$\frac{\partial y_i}{\partial S_i} = S_i{}' \tag{3.18}$$

$$\frac{\partial S_i}{\partial \omega_{ji}} = y_j \tag{3.19}$$

$$\Delta\omega_{ji} = -\eta \frac{\partial E}{\partial \omega_{ji}} = \eta\,(d_i - y_i)\, y_j\, S_i{}' \tag{3.20}$$

In the second case, for a hidden layer node j, the weight update rule can be expressed as (3.26), if the (3.22), (3.23), (3.24) and (3.25) are put in the equation (3.21).

$$\Delta\omega_{ji} = -\eta \frac{\partial E}{\partial \omega_{kj}} = -\eta \frac{\partial E}{\partial y_m} \frac{\partial y_m}{\partial S_m} \frac{\partial S_m}{\partial y_j} \frac{\partial y_j}{\partial S_j} \frac{\partial S_j}{\partial \omega_{kj}} \tag{3.21}$$

$$\frac{\partial E}{\partial y_m} = -(d_m - y_m) \tag{3.22}$$

$$\frac{\partial y_m}{\partial S_m} = S_m{}' \text{ and } \frac{\partial y_j}{\partial S_j} = S_j{}' \tag{3.23}$$

$$\frac{\partial S_m}{\partial y_j} = \omega_{jm} \tag{3.24}$$

$$\frac{\partial S_j}{\partial \omega_{kj}} = y_k \tag{3.25}$$

$$\Delta\omega_{kj} = -\eta \frac{\partial E}{\partial \omega_{ji}} = \eta y_k S_j{}' \sum_m (d_m - y_m)\omega_{jm}\, S_m{}' \tag{3.26}$$

3.3 SUPPORT VECTOR MACHINE

Support Vector Machines (SVM's) are the learning systems that aim to find the largest separation margin within an observed data set. Their learning algorithm is based on optimization theory that implements a learning bias derived from statistical learning theory. Support vector machine classifiers are first introduced by Vapnik (Cortes & Vapnik, 1995) and it is a principle and a powerful method with its usage over many classification tasks.

For binary classification implementing a SVM means finding the variables ω and b so that the training data can be described by:

$$x_i \cdot \omega + b \geq +1 \qquad \text{for,} \qquad y_i = +1 \tag{3.27}$$

$$x_i \cdot \omega + b \leq -1 \qquad \text{for,} \qquad y_i = -1 \tag{3.28}$$

$$y_i(x_i \cdot \omega + b) - 1 \geq 0 \qquad \forall_i \tag{3.29}$$

The points that are closest to the separating hyper-plane are considered as support vectors x_s. If two planes H_1 and H_2 are the outer borders of the hyper plane where the support vectors lie on, then these planes can also be described by the below equations.

$$x_i \cdot \omega + b = +1 \qquad \text{for,} \qquad H_1 \tag{3.30}$$

$$x_i \cdot \omega + b = -1 \qquad \text{for,} \qquad H_2 \tag{3.31}$$

If d_1 and d_2 are defined as the distances from $H_1$and H_2 to the hyper plane respectively, this two distances must be equal. So $d_1 + d_2$ distance gives us the SVM's margin. In order to keep the hyper planes as far as from the Support Vectors possible, this margin should be maximized. The margin is equal to $\frac{1}{\|\omega\|}$ and maximizing it is equal to finding:

$$min\|\omega\| \quad \text{s.t.} \quad y_i(x_i \cdot \omega + b) - 1 \geq 0 \quad \forall_i$$

Minimizing $\|\omega\|$ is equivalent to minimizing $\frac{1}{2}\|\omega\|^2$ and the use of this term makes it possible to perform quadratic programming (QP) optimization. We therefor need to find:

$$min\frac{1}{2}\|\omega\|^2 \qquad \text{such that,} \qquad y_i(x_i \cdot \omega + b) - 1 \geq 0 \qquad \forall_i \tag{3.32}$$

To handle the constraints in the minimization, we need to allocate them positive Langrange multipliers, denoted by α.

$$L_p \equiv \frac{1}{2}\|\omega\|^2 - \alpha[y_i(x_i \cdot \omega + b) - 1\ \forall_i] \tag{3.33}$$

$$L_p \equiv \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{L} \alpha_i[y_i(x_i \cdot \omega + b) - 1] \tag{3.34}$$

$$L_p \equiv \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{L} \alpha_i y_i(x_i \cdot \omega + b) + \sum_{i=1}^{L} \alpha_i \tag{3.35}$$

As the next step, the coefficients ω, b and α should be found which minimizes the equation (3.35). This can be achieved by taking the differential equation of L_p with respect to ω and b.

$$\frac{\partial L_p}{\partial \omega} = 0 \quad \Longrightarrow \quad \omega = \sum_{i=1}^{L} \alpha_i y_i x_i \tag{3.36}$$

$$\frac{\partial L_p}{\partial b} = 0 \quad \Longrightarrow \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.37}$$

Putting (3.36) and (3.37) into the equation (3.35) gives a new formula (3.38) that should be maximized which is dependent on α rather that ω and b.

$$L_D \equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad \text{for,} \quad \alpha_i \geq 0 \ \forall_i, \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.38}$$

$$\equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i H_{i,j} \alpha_j \quad \text{where,} \quad H_{i,j} \equiv y_i y_j x_i \cdot x_j \tag{3.39}$$

$$\equiv \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T H \alpha \quad \text{for,} \quad \alpha_i \geq 0 \ \forall_i, \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.40}$$

In this new formulation schema L_D is referred to as the Dual form of the Primary L_p. Dual form requires only the dot product calculation of each input vector x_i . For maximizing L_D instead of minimizing L_p we need to find:

$$\max_{\alpha} \left[\sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T H \alpha \right] \quad \text{for,} \quad \alpha_i \geq 0 \ \forall_i \quad \text{and} \quad \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{3.41}$$

This is called a convex quadratic optimization problem. This problem can be solved with a QP solver to derive the α. Then by putting the α in the equation (3.36) we can get the ω. The next step is the calculation of the second coefficient b.

Any data point satisfying (3.37) will simply have the form $y_s(x_s \cdot \omega + b) = 1$. Taking ω from (3.36) and putting it in equation $y_s(x_s \cdot \omega + b) = 1$ gives:

$$y_s \left(\sum_{m \in S} \alpha_m y_m x_m \cdot x_s + b \right) = 1$$

In the equation S denotes to the set of indices of the Support Vectors. S is determined by finding the indices i where $\alpha_i > 0$. Multiplying through by y_s and then using ${y_s}^2 = 1$ from (3.27) and (3.28) we can get the b:

$$ {y_s}^2 \left(\sum_{m \in S} \alpha_m y_m x_m \cdot x_s + b \right) = y_s $$

$$ b = y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s $$

Instead of using an arbitrary Support Vector x_s, an average over all of the Support Vectors in S can be taken as:

$$ b = \frac{1}{N_s} \sum_{s \in S} \left(y_s - \sum_{m \in S} \alpha_m y_m x_m \cdot x_s \right) \tag{3.42} $$

So with the above equation the variables ω and b define the optimum separating hyper plane (Fletcher, 2009) and each new point x' can be classified by evaluating the equation $y' = sgn(\omega \cdot x' + b)$.

3.4 NAIVE BAYES CLASSIFIERS

To understand the underlying dynamics of Bayesian learning machines, first the Bayes theorem should be understood. Basically given a set of data $x = \{x_1, \ldots, x_n\}$ the learning task is to uncover properties of the distribution from which the observed set is constructed. Bayes rule is a technique to estimate the likelihood of a property given the set of data as evidence, supposing x_i is an observable event, h_1 and h_2 are hypothesizes that may occur where $h_1 \wedge h_2$ cannot occur together. Then Bayes rule states:

$$ P(h_1|x_i) = \frac{P(x_i|h_1)P(h_1)}{P(x_i|h_1)P(h_1) + P(x_i|h_2)P(h_2)} \tag{3.43} $$

Here $P(h_1|x_i)$ is called posterior probability, while $P(h_1)$ is the prior probability associated with hypothesis h_1. $P(x_i)$ is the probability of the occurrence of data value x_i and $P(x_i|h_1)$ is the conditional probability that, given a hypothesis, the observation satisfies it. When there are m different hypotheses then we have (3.44) and in the most common form it is equal to the (3.45).

$$P(x_i) = \sum_{j}^{m} P(x_i|h_j)P(h_j) \tag{3.44}$$

$$P(h_1|x_i) = \frac{P(x_i|h_1)P(h_1)}{P(x_i)} \tag{3.45}$$

A naive Bayes classifier is a linear classifier which is based on applying the Bayes Rule of conditional probability among the features of an observed sample. In this assumption, the presence or absence of a feature doesn't affect or is not affected by the presence or absence of other features and the contribution of each feature to the classification result is equal. Because of its naive assumptive nature, these classifiers need only the mean and variance of the variables to estimate the parameters needed for the classification. For that reason, since the learning schema is supervised, in the training phase a small amount of data is mostly enough for the classifier.

When classifying a tuple, the conditional and prior probabilities generated from training set are used to make the classification (Dunham, 2002, pp. 92-97). Supposing that tuple t_i has p independent feature values $\{x_{i1}, x_{i2}, \ldots., x_{ip}\}$, since we know $P(x_{ik}|C_j)$, for each class C_j and feature x_{ik}, then we can estimate $P(t_i|C_j)$ by:

$$P(t_i|C_j) = \prod_{k=1}^{p} P(x_{ik}|C_j) \tag{3.46}$$

The naive Bayes approach has its own advantages. First it is easy to use. Second only one scan of the training data is required where sometimes there are cases that attributes are not independent meaning to unsatisfactory results for Bayes classifiers.

3.5 ENSEMBLE ALGORITHM

The ensemble learning is a machine learning algorithm that aims to combine "multiple base-learners" to learn the environment from their collective reasoning. The ensemble algorithm is not a standalone title in the machine learning family as it inherits the fundamental concepts of supervised learning. Since the algorithm is a type of supervised learning the prediction mechanism, which is the agent, should first be trained with a training set and then it can be used to make reasoning for a given new case. At that point, the main difference with the classical supervised mechanism is that, the trained agent is not a singular entity; rather it is a collection of the same or different base-classifiers. In such an environment a strong learner can be constructed from many relatively weak learners. There can be many methods (Alpaydin, 2009, pp. 419-442) or different ideas for combining multiple learners. Most fundamental methods can be listed as; random subspace method, bagging (aka Bootstrapping), boosting and cascading.

3.5.1 Random Subspace Method

If the input set has one multiple feature vector per sample, then by choosing random subsets of the vector features we can train multiple learners with different feature sets. Thus each "same base-learner" can look at the same problem from different points where this method is called as the random subspace or Ho's method.

3.5.2 Bagging (Bootstrapping)

For a given input set if we subgroup the input vectors rather than constructing features subgroups, then this method is called as the bagging or bootstrapping.

3.5.3 Boosting

In boosting algorithm, the collection of base-learners is built upon the misclassification results of former collections where in bagging this is done by chance.

3.6 DECISION TREE BASED CLASSIFIERS

Decision tree (DT) based learning approach is mostly used in classification tasks. Within this technique the main aim is the construction of a tree structure to model the classification process. As a simple notation, decision trees are the hypotheses that iteratively apply a divide and conquer technique to split the problem search space into smaller subsets. The root node of the tree is at the top where the sub nodes are connected to it by the arcs. The root and each internal node in the tree are labeled with a question and the arcs represent the answers to that question. The connections continue until the leaf nodes where the final prediction to the problem is represented. A decision tree building algorithm (Dunham, 2002, pp. 86-89) can be given as:

Input : D $// Training\ data$
Output : T $// Decision\ tree$

DT-Build algorithm :

$T = \emptyset$;
$Determine\ best\ split\ criterion$;
$T = Create\ root\ node\ and\ label\ with\ splitting\ atrribute$;
$T = Add\ arc\ to\ root\ node\ for\ each\ split\ predicate\ and\ label$;

for $each$ "arc" ***do*** :
$D = splitting\ predicate\ applied\ D$;
if ($stopping\ point\ reached\ for\ this\ part$)
$T' = Create\ leaf\ node\ and\ label\ with\ appropriate\ class$;
else
$T' = DTBuild(D)$;
$T = Add\ T'\ to\ arc$;
end for.

Exit algorithm.

If the size of the training data is q, and the number of attributes is h, then the time complexity of DT algorithm to build the tree can be regarded as $O(hq \log q)$.

3.6.1 Random Forest

The Random Forest (RF) algorithm was first developed by Leo Breiman and Adele Cutler as a supervised ensemble classification algorithm. The word ‘forest’ is used in the name of the algorithm to state that rather than using a single decision tree, the random forest algorithm uses a group of binary decision tree’s that can be called as a forest. The word ‘random’ indicates that this forest is constructed from the random subsets of the main training set with replacement, more formally, with bootstrapping. In a practical application upon the training set size, a random forest can be built from ten to thousands of discrete decision trees. Most of the nonlinear classification tasks can be achieved by the random forest classifiers. The random forest algorithm is most suitable for the cases where the training set is too large or the number of the variables for each vector is too many.

A random forest algorithm is consisted of two distinct phases. In the first phase the forest structure is constructed from the random subsets of the labeled main set. This phase can be named as the training phase. For a classification task let’s assume that we have a population of N vectors in hand and aim to construct a random forest with ‘n’ number of decision trees. To construct the forest structure first we randomly choose ‘n’ number of subsets of the main dataset. While the subsets are being chosen, the replacement is applied for each new step. That simply means that the subsets are free to be similar. Thus each decision tree in the forest is grown based on the randomly chosen n subsets. As a second parameter, the number of variables can be restricted to a constant number. Thus the variables between the subsets can be different while keeping them same within each subset. For both cases the best splitting criteria of the district trees is determined by the randomly chosen subset vectors. While growing the trees, no pruning is done to the decision trees to ensure each tree can fully grow.

In the second phase, the constructed forest structure is used to predict an unlabeled vector’s class. This unlabeled vector is such a one that the forest has never seen before. This phase can be simply named as the testing phase. The unlabeled vector is classified with each decision tree in the forest. The classification result of each district tree is called as the vote. Thus rather than a single decision tree, the final classification is done upon the majority of the votes by the forest.

3.7 DISTANCE BASED ALGORITHMS

The main idea behind the distance based classification algorithms is that, a pattern within a class is more alike to those which are in the same class rather than to else in other classes. So if a distance metric is defined between two points then a dissimilarity measure can be found for each given query points.

Mostly two distance metrics are used for determining the distance between the two multidimensional points, namely, Euclidean distance (3.47) and the Manhattan distance (3.48). For given two vectors $a_i = \langle a_{i1}, \ldots\ldots, a_{ik} \rangle$ and $a_j = \langle a_{j1}, \ldots\ldots, a_{jk} \rangle$ Euclidean and Manhattan distances can be given as:

$$distEuclidean(a_i, a_j) = \sqrt{\sum_{h=1}^{k} (a_{ih} - a_{jh})^2} \tag{3.47}$$

$$distManhattan(a_i, a_j) = \sum_{h=1}^{k} |a_{ih} - a_{jh}| \tag{3.48}$$

With the given proximity metrics, the basic classification task for a distance based algorithm is first finding the distances from a given point to all possible destinations and then assigning the given point to the closest group.

3.7.1 k-NN Algorithm

The k-Nearest Neighbor (kNN) algorithm is a nonlinear classification algorithm which mainly aims to classify a given point by regarding the distances from that point to all other neighbors. The main criterion for the classification is the majority of the neighbor labels where the letter “k” is a user defined constant denoting to the number of the neighbors that will be taken into account while making the classification. From the point of applicability, the k-Nearest Neighbor algorithm is considered as the simplest among all other machine learning algorithms.

The algorithm is mainly consisted of two phases which are the training and the test phases. The training examples are multidimensional vectors and they also contain the class labels. In the training phase all given training set is mapped into the memory and no any other process is done on this mapped data structure. Thus it is clear that the training phase doesn't need any weight update or other mathematical procedures where most of the work is done in the test phase. Since the mapping of the training set into the memory doesn't need any special effort, this algorithm is mostly called as lazy learning in machine learning community.

After the mapping process of the training examples is complete, the test phase starts for the classification of the target vectors. As the first step of the test phase, a distance table is constructed where the table contains; the indices of all mapped vectors, distance values from given vector to each vector and the vector labels (each in separate columns). As the second step the table is sorted up to the distance values column in an ascending order. Thus the topmost distance is the minimum distance which indicates the closest member of the mapped set to the tested vector. As the last step of the test phase a user defined constant value "k" is used to pick the k amount of closest neighbors for the classification. The constant k can take any values between one and the number of the vectors in the memory. The result of the classification is determined by the majority of the k nearest neighbor labels. If the k is one, then the classification result is the label of the topmost vector in the table. In most applications the classification result is directly affected by the constant k and choosing a high value for the k means more generalization.

CHAPTER 4

DIGITAL SIGNAL PROCESSING METHODS

Digital signal processing (DSP) is mainly used for time-frequency transformations. In time domain a periodic sinusoidal signal has a repeating “shape pattern”. So within a period the signal is represented with multiple output values according to time axis input values. In the “frequency domain” the same signal can simply be represented with a constant frequency value regardless of the signal duration. The constant frequency value shows the number of cycles the periodic signal achieved during its active state. Thus there can be a way to represent multiple overlapping time-domain signals separately in the frequency domain with the use of time-frequency transformations as long as the overlapping signal frequencies are different.

As being one of the most used DSP algorithms, this chapter basically aims to review the theory and the inner dynamics of the Fourier analysis process with the technical details of the unique computable Fast Fourier Transform (FFT) algorithm. Besides the FFT algorithm the theory of all other non-computable Fourier family members are also reviewed within the corresponding titles.

4.1 FOURIER ANALYSIS OF SIGNALS

The Fourier analysis is one of the most used digital signal processing (DSP) techniques for analyzing the signals in both time and frequency domains and it is named after Jean Baptiste Joseph Fourier (1768-1830), a French mathematician and physicist. The method is first proposed for the use of sinusoids to represent temperature distributions in heat propagation where today the theory is applied onto many different branches of engineering and science. The main aim of the Fourier analysis is defining

the periodic waveforms in terms of trigonometric functions (sine and cosine) and transforming them in-between the time and frequency domains.

In the Fourier analysis a signal is considered as a type of four forms (Smith , 1997, pp. 141-146). From the point of sampling, a signal can be continuous or discrete where this signal can be periodic or aperiodic in terms of the repeated shape. Each of the four possibilities is under the interest of Fourier analysis and represented with different titles in the Fourier family. This organization can be viewed as follows:

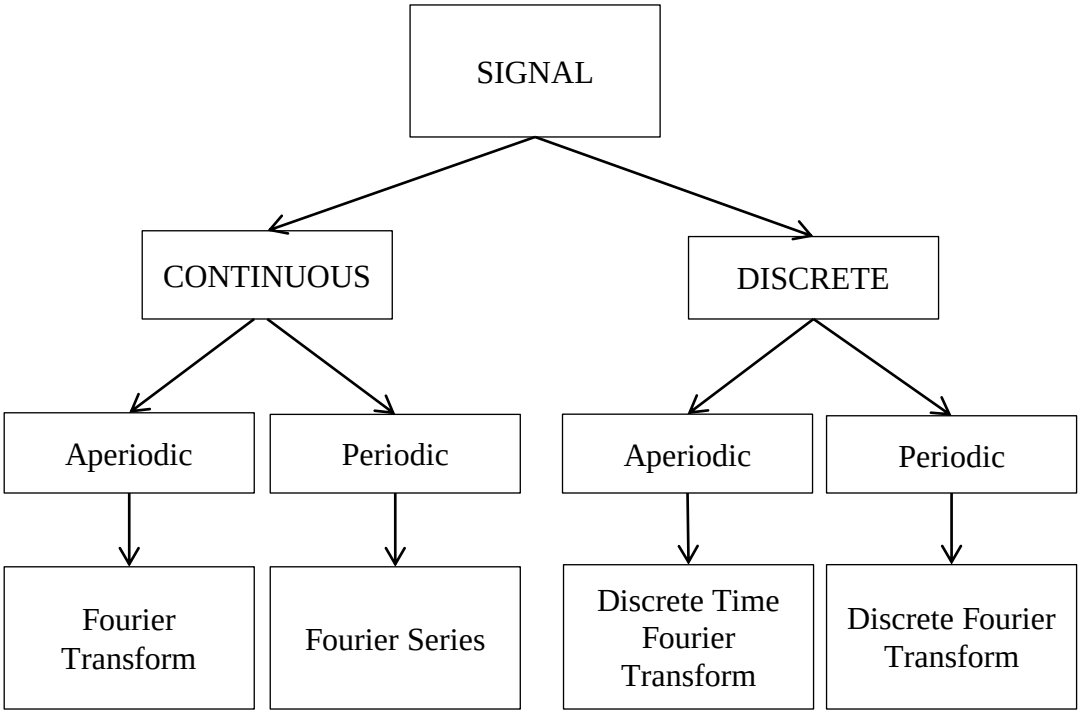


Figure 4.1 The basic Fourier family tree.

4.1.1 Continuous and Discrete Time Signals

To better understand the Fourier analysis, first the signals that are subject to the Fourier process should be understood. Signal processing deals with two kinds of time varying signals which are named as analogue and digital. Thus the signal processing is divided into two categories, namely, “analogue signal processing” and “digital signal processing”. In a continuous (analogue) signal, the signal is a function of infinite time

variants. The waveform is continuous in time and can take on a continuous range of amplitude values. The discrete-time signals are a special form of the continuous signals where the independent time variable is quantized so that the value of the signal at discrete instants in time is known. As a result of the quantization the discrete time signals are not represented by continuous waveforms where the representation is by discrete sequenced values.

4.1.2 Fourier Series

The Fourier series is used for expanding a continuous periodic signal into a linear combination of sine and cosine waves. The calculation or the study of Fourier series is also known as the harmonic analysis. A similar idea in mathematics is the Taylor series where a function is expanded into infinite series of powers. If a function f(x), with period 2π, is absolutely integrable on [−π, π] and continuous, then the Fourier series of the function f(x) is as follows:

$$f(x) = \frac{1}{2}a_0 + \sum_{n=1}^{\infty} a_n \cos(nx) + \sum_{n=1}^{\infty} b_n \sin(nx) \tag{4.1}$$

$$where, \quad a_0 = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\,dx \tag{4.2}$$

$$a_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\,cos(nx)\,dx \tag{4.3}$$

$$b_n = \frac{1}{\pi}\int_{-\pi}^{\pi} f(x)\,sin(nx)\,dx \tag{4.4}$$

4.1.3 Fourier Transform (CTFT)

Fourier transform is the process of converting a signal from time domain into the frequency domain where the inverse Fourier transform expresses a frequency domain function in the time domain. Since the transform is applied onto the continuous signals, this method is also called as the continuous time Fourier-transform (CTFT). The method can be regarded as the decomposition of non-periodic functions into Fourier components and it is a generalized form of complex Fourier series.

$$F(k) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i k x}\, dx \tag{4.5}$$

$$f(x) = \int_{-\infty}^{\infty} F(k)\, e^{2\pi i k x}\, dk \tag{4.6}$$

In the above equations, the forward Fourier transform (4.5) is denoted by $F(k)$, where the inverse Fourier-transform (4.6) is denoted by $f(x)$. Fourier transform is mostly used in solving differential equations since it is closely related to Laplace transformation.

4.1.4 Discrete Time Fourier Transform (DTFT)

In Fourier analysis, the discrete time Fourier transform (DTFT) is a specific type of transformation that is used for transforming a 'discrete and aperiodic' time domain signal in a continuous frequency form. Since the computers cannot handle continuous frequency data, DTFT is practically not computable but it provides a theoretical basis for the Z transform. If $x(n)$ is absolutely summable, then the DTFT is:

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j\omega n} \qquad where, \qquad \sum_{n=-\infty}^{\infty} |x(n)| < \infty \tag{4.7}$$

4.1.5 Discrete Fourier Transform (DFT)

The discrete Fourier transform (DFT) is used to transform the “periodic and discrete time signals” into the frequency domain. Since the DFT requires its inputs to be discrete and finite, it is the unique “computable algorithm” where the other members of the Fourier analysis are non-computable due to their infinite input needs. Literally, given the time domain signal, the process of calculating the frequency domain is called decomposition, analysis, forward DFT or simply DFT, where the calculation of the time domain from a given frequency domain is called synthesis or inverse-DFT.

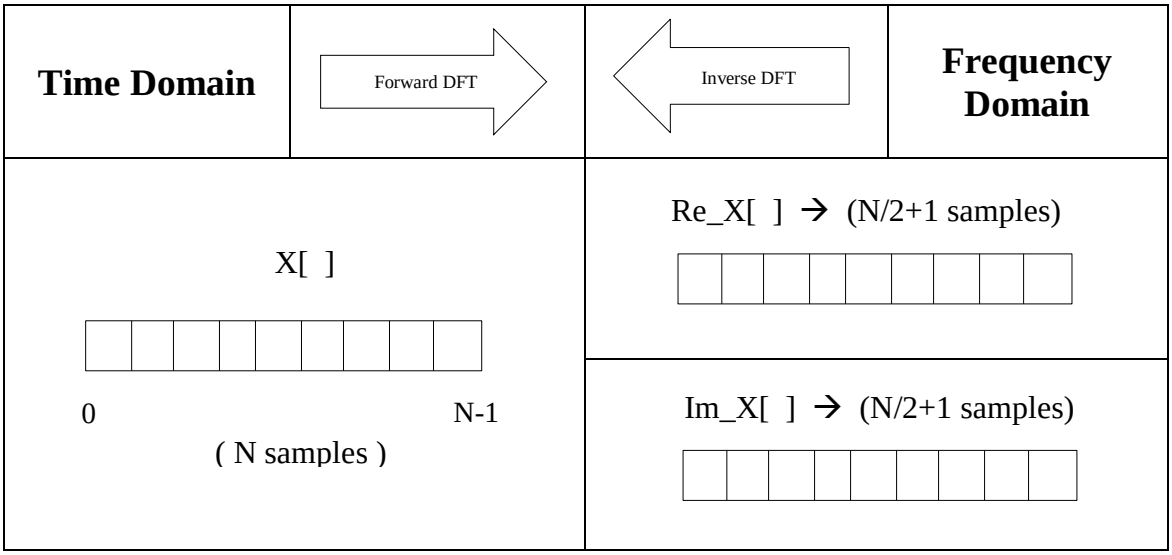


Figure 4.2 The schema of the forward and inverse DFT.

There are two types of DFT from the view of representation, namely, real-DFT and complex-DFT. The real-DFT changes an N point input signal into two different (N/2+1) point output signals where the complex-DFT changes a set of N complex points into another set of N complex points. For the real-DFT the input signal x[] is consisted of the discrete values of a time signal chunk being decomposed, while the two output signals contain the amplitudes of the sine and cosine waves. This two output signals are called as the real and imaginary parts of input x[]. If the input to a DFT is N point time signal, then it is clear that the output is N+2 points in total. The 2 extra points in the output are the values of Im_X[0] and Im_X[N/2] which contains no information and are always zero.

4.1.5.1 Rectangular Notation

For a given time signal, the DFT can be computed by finding the amplitudes of the sine and cosine waves which individually represent the imaginary and the real parts of the frequency domain. Let's assume x[i] as the time domain signal to be analyzed, where the index 'i' takes values from 0 to N-1 (N points). If the frequency domain components being calculated are ReX[k] and ImX[k], where the index 'k' takes values from 0 to N/2 (N/2+1 points), then the analysis equation for DFT can be given as:

$$X[k] = ReX[k] + j\, ImX[k] \quad where, \tag{4.8}$$

$$ReX[k] = \sum_{i=0}^{N-1} x[i] \cos\left(\frac{2\pi ki}{N}\right) \tag{4.9}$$

$$ImX[k] = -\sum_{i=0}^{N-1} x[i] \sin\left(\frac{2\pi ki}{N}\right) \tag{4.10}$$

In both equations (4.9) and (4.10), the sine and cosine terms are called the DFT basis functions where both equations are in the rectangular form. The analysis equations can be interpreted as calculating the frequency domain by multiplying the time domain signal with the basis-functions and collectively adding the resulting points, where this process is simply called the correlation. As a property of the DFT, if any two basis functions are multiplied, the sum of the multiplication should always result to zero. This property is called as orthogonality and the basis functions that meet this property are called as orthogonal. The inverse-DFT process can be formulated as follows:

$$x[i] = \sum_{k=0}^{N/2} Re\bar{X}\,[k] \cos\left(\frac{2\pi ki}{N}\right) + \sum_{k=0}^{N/2} Im\bar{X}\,[k] \sin\left(\frac{2\pi ki}{N}\right) \quad where, \tag{4.11}$$

$$Re\bar{X}[k] = \frac{ReX[k]}{N/2} \quad and, \quad Im\bar{X}[k] = -\frac{ImX[k]}{N/2} \tag{4.12}$$

$$Re\bar{X}[0] = \frac{ReX[0]}{N} \quad and, \quad Re\bar{X}[N/2] = \frac{ReX[N/2]}{N} \tag{4.13}$$

4.1.5.2 Polar Notation

As an alternative to the rectangular notation, where the frequency domain is a group of amplitudes of cosine and sine waves, the DFT equations can be described in polar notation. The process of rectangular to polar transformation simply aims to represent the frequency domain components by a vector (MagX) with an angle (theta), instead of two discrete vectors (ReX) and (ImX).

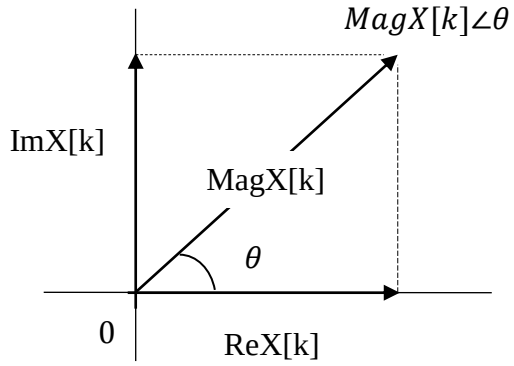


Figure 4.3 Polar notation geometric representation.

In this new notation, the real and the imaginary components of the DFT equations can be replaced with the magnitude and phase components. The polar form transformation equations are as follows:

$$MagX[k] = \sqrt{(ReX[k]^2 + ImX[k]^2)} \tag{4.14}$$

$$PhaseX[k] = \theta = tanh^{-1}\left(\frac{ImX[k]}{ReX[k]}\right) \tag{4.15}$$

The both rectangular and the polar representations contain exactly the same information. From the view of computability, the rectangular notation is more practical and efficient when computing a DFT algorithm where the polar notation is used to visualize the DFT results just because a frequency domain signal cannot be easily understood by just looking at the real and imaginary parts.

4.1.5.3 Complex Notation

Although the DFT equations are fine in rectangular notation it is possible to simplify the equations in the complex notation with the Euler's identity, such as:

$$e^{j\varphi} = cos\,\varphi + j\ sin\,\varphi \tag{4.16}$$

Complex numbers represent points in a two dimensional plane where the referencing to the point is done with the use of two distinct axes. The horizontal axis is the real-axis and the vertical axis is called as imaginary axis. For the given Euler's equation (4.16), the geometric representation of the complex notation is as follows:

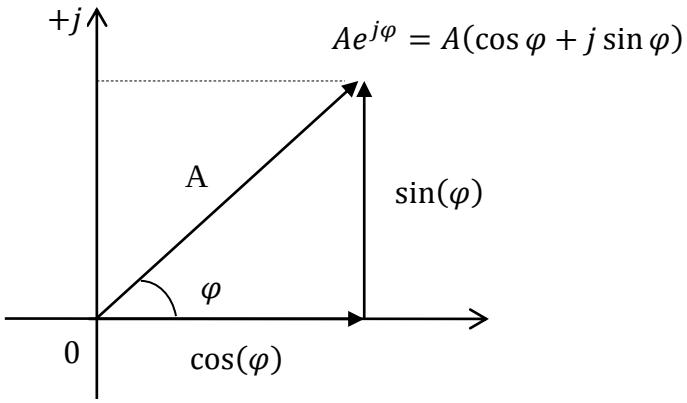


Figure 4.4 Complex notation geometric representation.

Regarding the Euler's identity (4.16), the forward-DFT (4.17) and the reverse-DFT (4.18) equations can be represented in complex notation as follows:

$$X[k] = \sum_{n=0}^{N-1} x[n]\, e^{\frac{-j2\pi kn}{N}} \qquad 0 \le k \le N-1 \tag{4.17}$$

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{\frac{j2\pi kn}{N}} \qquad 0 \le n \le N-1 \tag{4.18}$$

4.2 FAST FOURIER TRANSFORM (FFT)

In the Fourier family the unique computable algorithm is the discrete Fourier transform (DFT). If the code structure of a DFT implementation is analyzed it is visible that for calculating the frequency domain components of an N point complex time domain signal, a block of a nested "for-loop" is needed where the inner and outer for-loops would both run through an index of size N. As the time complexity the total computation is equal to $O(N^2)$. Hence for processing a time signal with 1024 points, a total of $1024^2 = 1{,}048{,}576 \cong 10^6$ calculations are needed. Since the amount of the calculations increase exponentially depending on the input size, the real time applications need more resources. To solve this problem, some improved algorithms were proposed by the researchers.

One of the most fundamental algorithms to speed up the DFT computation was proposed in 1965 by Cooley et al., the Cooley-Tukey Algorithm (Cooley & Tukey, 1965), with its famous name fast Fourier transform (FFT). The FFT is simply a divide and conquer algorithm where the N point input signal is recursively divided into N/2 size sub-units (even-odd parts) in an interlaced manner. Thus the complexity of the DFT process is lessened from $O(N^2)$ to $O(N \log N)$. The FFT algorithm is a two-step algorithm. In the first step, the N point time domain input is repeatedly divided into two groups as being odd and even parts. At the end of the $(\log_2 N)^{th}$ step, the N point input signal is decomposed into N signals where each signal consists of a single point. The process of 8 point signal decomposition is as follows:

Table 4.1 Input decomposition of an 8 point signal.

Decomposition of The Signal								Steps
0, 1, 2, 3, 4, 5, 6, 7								$input\ signal$
0, 2, 4, 6				1, 3, 5, 7				$1^{st}\ step$
0, 4		2, 6		1, 5		3, 7		$2^{nd}\ step$
0	4	2	6	1	5	3	7	$3^{th}\ step$

4.2.1 Bit Reversal Sorting Algorithm

The decomposition process is simply the reordering of the points in an N point signal at the end of a $(\log_2 N)$ steps iterative division. This process can be simplified with the bit reversal sorting algorithm.

Table 4.2 Bit reversal sorting of an 8 point signal.

Decimal Input	Binary	Reversed Binary	Decimal Output
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

In the bit reversal sorting algorithm, index of the each point is converted from the decimal form to the binary form. Then each binary number is simply reversed where for each signal the reversed numbers point to the new indexes in the decimal form. If both bit reversal sorting and recursive dividing methods are compared, the final sorted outputs are completely identical.

4.2.2 Frequency Synthesis Process

After the decomposition, or bit reversal sorting process, each 1 point signal is now a frequency spectrum. The common bug at that point is assuming these 1 point signals still to be a time domain signal. As the last step of the FFT algorithm, the N point frequency spectra should be combined. This synthesis process is a step by step process in such a way that in the first step the 8 x (1 point) spectra is synthesized into a 4 x (2 points) spectra. In the second step the 4 x (2 points) spectra is synthesized into a 2 x (4 points) spectra. Lastly in the third step the 2 x (4 points) spectra is synthesized into a 1 x (8 points) frequency spectrum. This is simply the reverse operation of decomposition.

In the FFT synthesis, the composition process can be represented in a structure which is called as the butterfly structure. The composition of an 8 x (1 point) spectra with the butterfly structure is as follows:

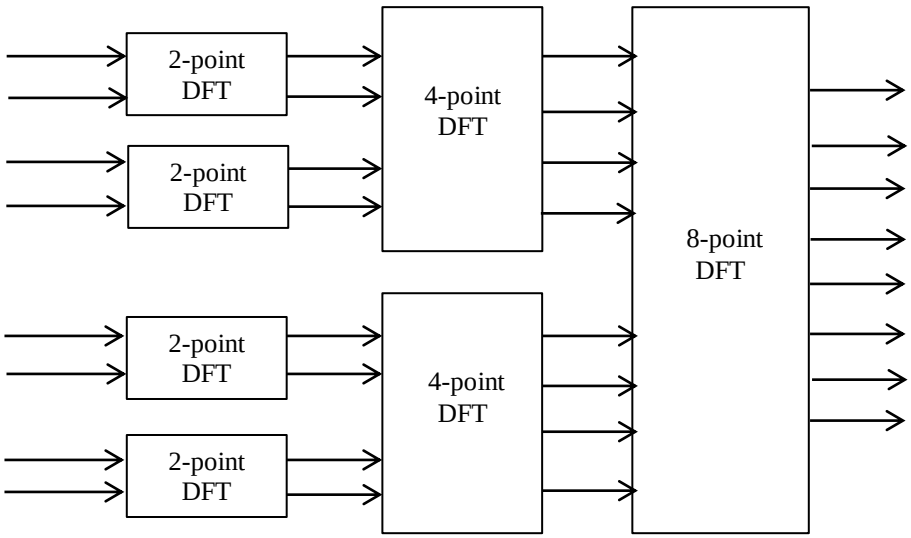


Figure 4.5 The 8 point signal synthesis.

In the synthesis process the input vector is “N x (1 point) spectra” which is in the sorted order, such that (0, 4, 2, 6, 1, 5, 3, 7), where the output vector is sorted as (0, 1, 2, 3, 4, 5, 6, 7). The inner diagram of the structure (Danielson-Lanczos Algorithm, 2012) is as follows:

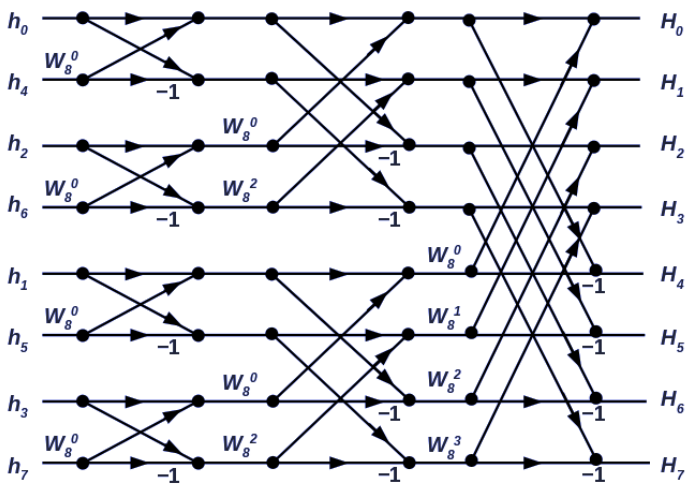


Figure 4.6 The 8 point synthesis inner diagram.

4.2.3 Single 2-Point DFT butterfly

As being the smallest butterfly unit and also the base for more complicated butterfly structures, the inner schema of a 2-point butterfly should be well understood. This unit accepts two inputs and produces two outputs. Each output is the correlation of the inputs with the inner twiddle factors. The basic structure of a 2-point DFT butterfly can be represented as follows:

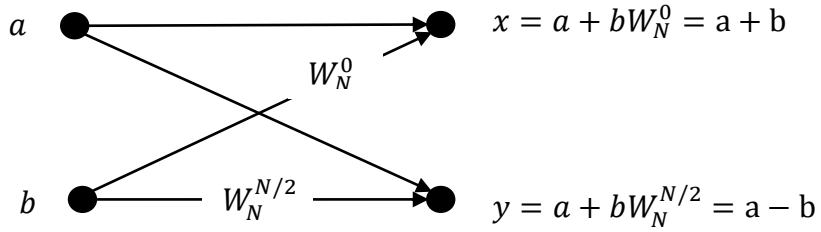


Figure 4.7 Single 2-point butterfly.

The twiddle factor is a multiplier in the butterfly structure and it is used for simplifying the DFT basis functions in the complex notation. The equation for the twiddle factor is as follows:

$$W_N^S = e^{-j2\pi\frac{S}{N}} \tag{4.19}$$

$$W_N^0 = e^{-j2\pi\frac{0}{N}} = e^0 = +1 \tag{4.20}$$

$$W_N^{N/2} = e^{-j2\pi\frac{N}{2N}} = e^{-j\pi} = cos\,\pi - j\,sin\,\pi = -1 \tag{4.21}$$

The 2-point DFT butterflies cannot be partitioned into smaller forms but by modifying the mathematical equation and the inner connections they can be represented in many ways (Lyons, 2011, pp. 141-154). The FFT algorithm used here is called decimation in time and if the input size is chosen as being the power of two, then the algorithm for the butterfly structure is called as the Radix-2 butterfly.

4.3 WINDOWING ALGORITHMS FOR FFT

Fourier analysis helps us a lot to understand a signal. The output of the analysis reports us the two major specifications of an unknown signal which is of our interest, namely, the amplitude and the frequency. So if we trust the FFT and re-plot the signal with the given amplitude and frequency measures, most of the time we can get the signal back in its original form. But in some cases FFT gives us some strange answers where the amplitude-frequency graph is full of unrelated components such as, a-true amplitude and frequency pair with some sidelobe containing huge amounts of unrelated amplitude-frequency pairs. This problem is called as the frequency leakage. The main reason of the frequency leakage is applying the FFT to a portion of a periodic signal where the starting and ending points of the signal are not same causing this problem to arise. And if the observed signal is a sum of multiple signals, the spectral leakage from a larger signal component may also swallow the other smaller signals making them too hard to identify. Windowing algorithms mainly aim to minimize the frequency leakage effects by lowering the amplitudes of the starting and ending points of a time domain signal chunk. Any discontinuities that are contained by the signal are also targeted. In this section some of the most used windowing algorithms and their mathematical representations will be presented.

4.3.1 Rectangular Window

While transforming a time domain signal into the frequency domain, there is no choice for not using a window over the target signal. So the rectangular window is the default window while taking a FFT. It is used implicitly. Rectangular window's magnitude is always 1 over the sample interval but zero for the points that are out of the window. The mathematical equation for the rectangular window is as follows:

$$w(n) = \begin{cases} 1, & 0 \leq n \leq N-1 \\ 0, & n < 0 \; or \; n \geq N \end{cases} \tag{4.22}$$

4.3.2 Hann Window

This function is called after the Julius Ferdinand von Hann, who was an Austrian meteorologist. The Hann window is sometimes called as the Hanning window. As an important property of the function, the start and end points are always zero. So the signals that were multiplied with this window also start and end with zero. The mathematical equation for the Hann window is as follows:

$$w(n) = \frac{1}{2} - \frac{1}{2}\cos\left(\frac{2\pi n}{N}\right) \qquad where, \qquad 0 \leq n \leq N-1 \tag{4.23}$$

The graphical representation for the Hann window is as follows:

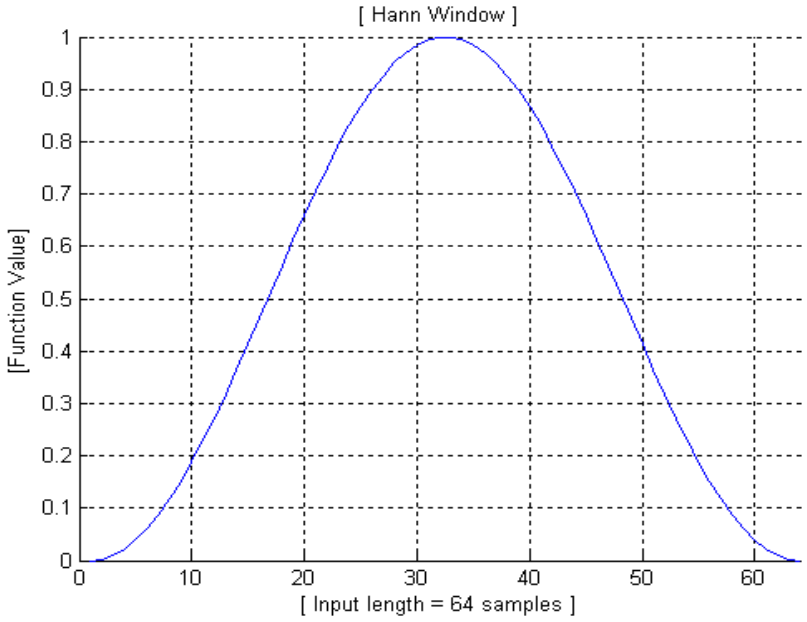


Figure 4.8 The Hann window.

4.3.3 Hamming Window

The Hamming window is in the same family with the Hann window. As a different property this function doesn't start and end with zero as default. So the data at the edge regions are not lost where they are just attenuated. For this reason the Hamming window is mostly preferred over Hann window. The basic mathematical equation for the Hamming window is as follows:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right) \qquad where, \qquad 0 \leq n \leq N-1 \qquad (4.24)$$

The graphical representation for the Hamming window can be given as follows:

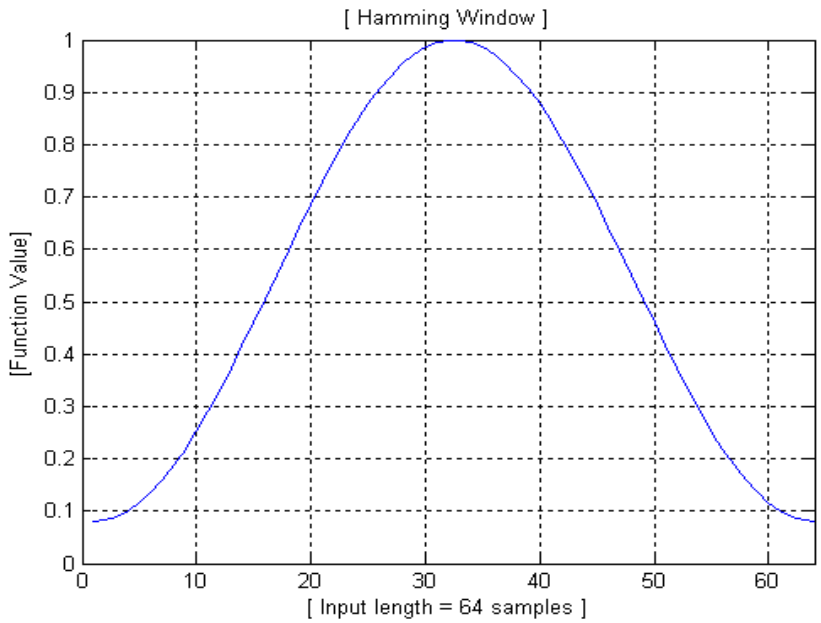


Figure 4.9 The Hamming window.

4.3.4 Blackman–Harris Window

The Blackman-Harris window is in the same family with the Hann and the Hamming windows. It has two types, namely, symmetric and periodic. In both cases the function has four constant coefficients where the odd coefficients are negative and the even coefficients are positive. As a common point with the Hann window, this function also start and end with zero. The mathematical equation of a 4-term and symmetric Blackman-Harris window is as follows:

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right) - a_3 \cos\left(\frac{6\pi n}{N-1}\right) \qquad (4.25)$$

$$where\ ; \quad 0 \le n \le N-1$$

$$a_0 = 0.35875$$

$$a_1 = 0.48829$$

$$a_2 = 0.14128$$

$$a_3 = 0.01168$$

The graphical representation for this window can be given as follows:

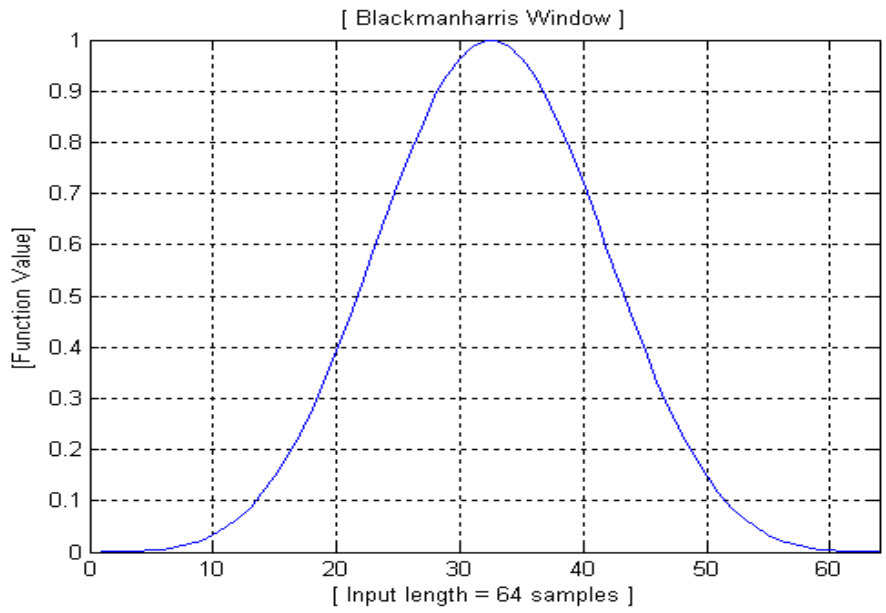


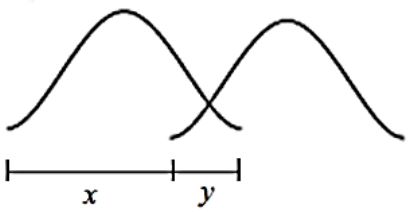
Figure 4.10 The Blackman-Harris window.

4.4 SHORT TIME FOURIER TRANSFORM (SFT)

To compute the frequency spectrum of a signal, the whole signal is represented to the FFT algorithm. But if the signal is too long or the district parts of the signal have different characteristics, then the signal can be partitioned into multiple smaller chunks. The idea of short time Fourier-transform is very parallel with the FFT algorithm where the computations are done for smaller units. After partitioning the signal, each chunk can be represented to the Fourier transform. This process is simply called as the short time Fourier transform (SFT). For computing the SFT, first a window type and then a window size should be determined. Then each constant length window is applied to the adjacent regions of the main signal and thus each window covers a different signal portion. The signal portion that is covered by a window occupies a time slice of the signal period. We compute the FFT for each time slices of a signal and so the SFT algorithm helps us to see the frequency components of each time slot where each user defined window length determines the time resolution.

4.4.1 Window Overlapping Technique

While computing the SFT, each window starts just after the former window finishes. This kind of windowing divides the signal into smaller and equal-size portions leading to no overlapping between the windows. But regarding the window type being applied, the information at the start and end points of each window may get lost or attenuated. The simple graphical representation of an overlapping enabled windowing process can be given as follows:



$$window\ length = x + y$$
$$step\ size = x$$
$$overlap\ ratio = \frac{y}{x+y}$$

Figure 4.11 Window overlapping technique.

The problem of zeroed regions can be denied by overlapping the adjacent window borders. Each window edge that was mutated by the windowing function can normally be included in the next window with the use of the overlapping technique. Since the window overlapping technique seems to be lifesaving, it has a cost of overlap ratio adjustment. If the overlap ratio is chosen high then the total window amount increases and that directly causes to an increase in total number of the calculations.

4.5 SPECTROGRAMS

Simply a spectrogram is the magnitude squared form of a short time Fourier-transform output. Spectrograms are used to visualize the frequency components and the spectral density of time varying signals in a 3 dimensional environment. They are also called as the voice finger prints, sonograms or voicegrams. The basic equation for the spectrograms is as follows;

$$spectogram(t,w) = |SFT(t,w)|^2 \tag{4.26}$$

While constructing the spectrograms, first the time domain signal is partitioned into smaller time chunks. Then a predefined windowing algorithm is applied to each time chunk. Thus the starting and the end points of each chunk are attenuated and the possible future frequency leakages are avoided. Then the FFT algorithm is applied to each window. The output of the FFT algorithm is a two dimensional graphics consisting of a horizontal “frequency” and a vertical “amplitude” axis. So far the process is nothing but the sort time Fourier transform. As the second step, each window’s FFT vertical “amplitude” axes are squared. Thus the differences between the low and high amplitude values are emphasized. And in the third step, all two dimensional squared FFT graphics are put next to each other to construct the third dimension what is called the time axis. So while computing the SFT, the number of used chunks determine the time resolution of the spectrograms.

CHAPTER 5

FEATURE EXTRACTION METHODOLOGY

In a digital environment the audio files are kept as a collection of discrete numerical values without regarding the overheads. Each chunk of an audio file represents the amplitude value of an audio signal at a given time point. The amount of the discrete points in a signal is proportional to the sampling rate of the digitization process. As an example if the sampling rate is 5000 samples per-second, which can also be denoted by 5 KHz, an audio clip with a 1 minute duration is equal to an array of 300,000 numerical discrete values where this amount is equal to a minimum memory space requirement of 2,4 MB's, assuming each value is kept in a 64bit-double primitive. At this point, an audio file can be regarded as a big mass when represented to a classifier for a classification task.

Basically the process of representing an audio file with minimum possible features (or dimensions) is called as the "audio feature extraction" process. The minimum length and maximum descriptive dimensions can be derived with two basic methods. In the first case the output vector of the extraction algorithm is a subset of the input vector values where this process can be regarded as a non-correlated attribute removal, feature selection or dimensionality reduction process. In the second case the input vector is simplified into a non-input type space where this process can be regarded as a mapping or signature extraction process. The feature extraction process not only simplifies or summarizes the sound vector, but also transforms it into a fixed size notation regardless of variable input length. If the audio files with different durations are summarized into a fixed size notation, then the comparison between them is also simpler for the machine learning algorithms.

5.1 MFCC

Mel frequency Cepstral coefficient (MFCC) algorithm is one of the most used feature extraction algorithms that summarizes an audio signal into a user defined, fixed-length coefficient vector which is called as Mel Frequency Cepstrum (MFC). In the first step of the MFCC algorithm, the audio signal is partitioned into multiple smaller chunks with an overlapping window where each window is weighted by a corresponding windowing function (Hann, Hamming etc.). This process is simply called as the windowing step. The need for the windowing step is the non-stationary characteristics of the speech signals. The windowing is used to have a small enough region of the signal where the spectral information of that region is a useful cue. After windowing step, the discrete Fourier transform (mostly FFT) is applied to each window. Until this point, the windowing step followed by FFT process is nothing but the short time Fourier-transform (SFT). At the end of the SFT process, the window content is represented in the raw frequency form. In this raw form each point is represented with two axes, namely, the vertical-Amplitude and the horizontal-Frequency axis. The frequency axis maximal value is the half value of the sampling rate. Both amplitude and the frequency axis values are linearly spaced in this raw form.

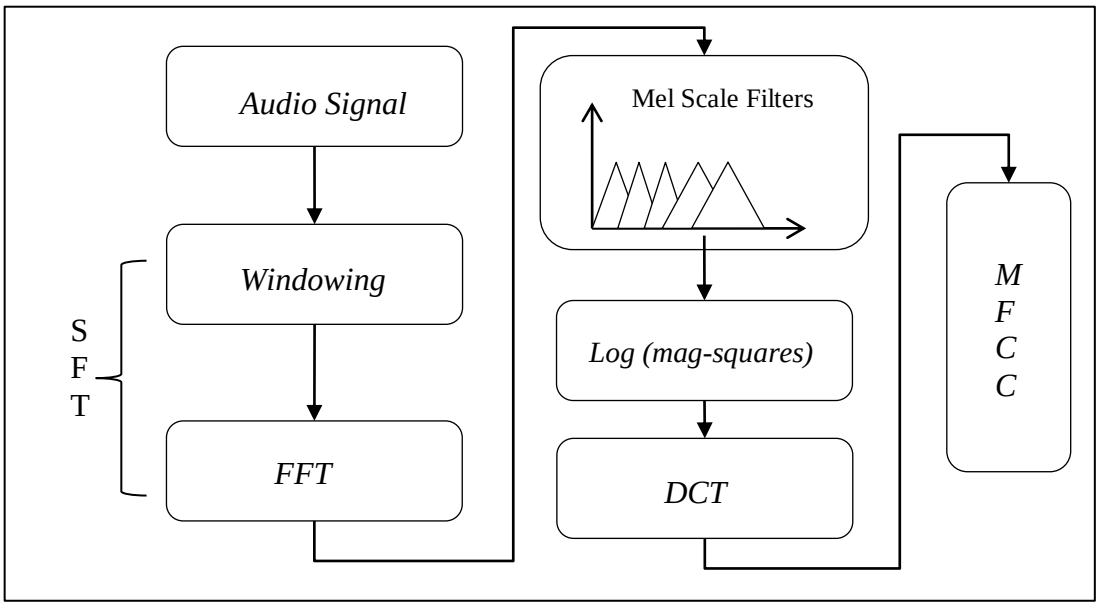


Figure 5.1 MFCC block diagram.

The human perception of frequency is non-linear as a result of the fact the human hearing is not equally sensitive to all frequency bands. Simply the sensitivity threshold for a human ear is accepted as 1000 Hz. The frequency values that are greater than the threshold are less sensitively percepted. By the given fact, the raw frequency axis (of the FFT process) is mapped onto the mel-scale, such that:

$$Mel(f) = 2595\ \log_{10}\left(1 + \frac{f}{700}\right) \tag{5.1}$$

The mel-scale is approximately linear below 1000 Hz and logarithmic above the 1000 Hz. After mel-scaling the raw spectrum, a Mel-filter Bank is applied to the mel-scaled spectrum.

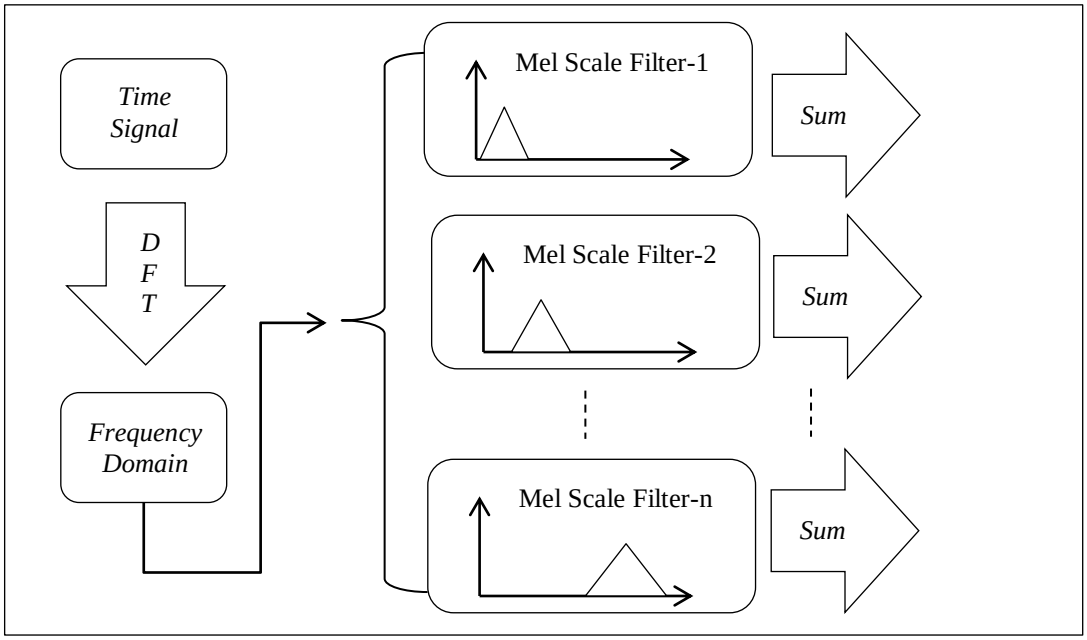


Figure 5.2 Mel-filter Bank application process.

The Mel-filter Bank is a triangular overlapping window structure. In the bank, filters are centered according to Mel-frequencies. The coverage of triangular filter is according to the mel-scale. Each filter output is the sum of its filtered spectral components. In a typical application the number of the filter-bank is a user defined integer value varying between (20 and 40) up to the design. After applying the mel-filter bank to the FFT outputs, simply a logarithm is applied to the magnitude squares of the filter bank outputs. This is just because the human response to the signal level is logarithmic. While talking to microphone the speaker may come closer to the recording device where that may cause to slight differences in the time domain input. Log computation makes frequency estimates less sensitive to slight variations in input. As the last step of the MFCC algorithm, the Discrete Cosine Transform (DCT) is computed from the outputs of the logarithm process. Since the logarithm of power spectrum is real and symmetric, the inverse-DFT and the DCT are identical to each other and can be used interchangeably. As the result, the product of the DCT process is the Cepstrum, with other words “the spectrum of the log of the spectrum”.

5.2 LPC

Liner prediction based speech analysis is mainly used for mapping a target sound sample to a finite length feature space. In linear prediction, the representation of the speech waveform is by the parameters of an all-pole model. These all-pole model parameters are simply called the linear predictive coefficients (LPC). For a given speech sample, LPC analyses basically tends to find an optimal fit to the speech spectrum where the process is done in time space rather than frequency domain. Each function can be estimated by LPC coefficients, meaning the value of a signal at a given time “t” is estimated with the linear combination of signal values in previous times. The LPC features are computed by autocorrelation or covariance methods. Given that, the LPC technique is equivalent to auto regressive (AR) speech signal modeling.

The first step in the LPC analyses is the Pre-emphasis step. In this step the signal is passed to a first order low-pass filter where the filter simply flattens the signal and makes it more stable to precision effects. Then the pre-emphasized signal is first framed and subjected to windowing process. The windowing process parses the signal into

multiple chunks with a user defined overlap. To avoid the discontinuities at the window borders, mostly the Hamming window is preferred. In the next step, each window is (*nth*-order) auto correlated where the autocorrelation is the correlation of a signal with itself. And lastly in the LPC analyses step, each frame of the autocorrelations is converted into an LPC parameter set with the use of recursive techniques such as Levinson/Durbin algorithm.

5.3 TIME DOMAIN FEATURES

The time domain features deals with the numeric values of a signal in the time domain rather than their properties in the frequency domain. With the use of time domain features, a signal or a numeric sequence can be defined with finite features regardless of its length. Most of the time domain features use the fundamental statistical methods such as; sum, average, mean etc., as their base methodology. The names and the short explanations of 10 Time Domain features that were used for the thesis study are as follows;

Table 5.1 Time domain feature names and definitions.

No	Feature Name	Short Definition
1	Spectral Centroid	The center of mass of the power spectrum.
2	Spectral Roll off Point	The fraction of bins in the power spectrum at which 85% of the power is at lower frequencies. This is a measure of the right-skewedness of the power spectrum.
3	Spectral Flux	A measure of the amount of spectral change in a signal. Found by calculating the change in the magnitude spectrum from frame to frame.
4	Compactness	A measure of the noisiness of a signal. Found by comparing the components of a window's magnitude spectrum with the magnitude spectrum of its neighboring windows.
5	Spectral Variability	The standard deviation of the magnitude spectrum. This is a measure of the variance of a signal's magnitude spectrum.
6	Root Mean Square	A measure of the power of a signal.
7	Fraction Of Low En. Win.	The fraction of the last 100 windows that has an RMS less than the mean RMS in the last 100 windows. This can indicate how much of a signal is quiet relative to the rest of the signal.
8	Zero Crossings	The number of times the waveform changed sign. An indication of frequency as well as noisiness.
9	Strongest Beat	The strongest beat in a signal, in beats per minute, found by finding the strongest bin in the beat histogram.
10	Beat Sum	The sum of all entries in the beat histogram. This is a good measure of the importance of regular beats in a signal.

5.4 PHONETIC FEATURES

For deriving the phonetic features of repository sounds, the PRAAT program (Praat, 2012) has been used. The program was developed at the University of Amsterdam and mainly used for analyzing the human speech properties. For analyzing the pulmonary sounds, all predefined program parameters has been re-studied. The 17 PRAAT features (Boersma & Weenink, 2001) used for the thesis are as follows;

Table 5.2 PRAAT phonetic feature names and definitions.

No	Feature Name	Short Definition
1	Mean pitch	The mean pitch value of the points within a specified time window.
2	Standard deviation	The standard deviation of the points within a specified time window.
3	Maximum pitch	The maximum pitch value of the points within a specified time window.
4	Minimum pitch	The minimum pitch value of the points within a specified time window.
5	Jitter (local)	The average absolute difference between consecutive periods, divided by the average.
6	Jitter (local, absolute)	The average absolute difference between consecutive periods, in seconds.
7	Jitter (rap)	The relative average perturbation, the average absolute difference between a period and the average of it and its two neighbors, divided by the average period.
8	Jitter (ppq5)	The five-point Period Perturbation Quotient, the average absolute difference between a period and the average of it and its four closest neighbors, divided by the average period.
9	Jitter (ddp)	The average absolute difference between consecutive differences between consecutive periods, divided by the average period.
10	Shimmer (local)	The average absolute difference between the amplitudes of consecutive periods, divided by the average amplitude.
11	Shimmer (local, dB)	The average absolute base-10 logarithm of the difference between the amplitudes of consecutive periods, multiplied by 20.
12	Shimmer (apq3)	The three-point Amplitude Perturbation Quotient, the average absolute difference between the amplitude of a period and the average of the amplitudes of its neighbors, divided by the average amplitude.
13	Shimmer (apq5)	The five-point Amplitude Perturbation Quotient, the average absolute difference between the amplitude of a period and the average of the amplitudes of it and its four closest neighbors, divided by the average amplitude.
14	Shimmer (apq11)	The 11-point Amplitude Perturbation Quotient, the average absolute difference between the amplitude of a period and the average of the amplitudes of it and its ten closest neighbors, divided by the average amplitude.
15	Shimmer (dda)	Average absolute difference between consecutive differences between the amplitudes of consecutive periods.
16	Mean NHR	Mean Noise-to-Harmonics Ratio value of the points within a specified time window.
17	Mean HNR	Mean Harmonics-to-Noise Ratio value of the points within a specified time window.

CHAPTER 6

STATISTICAL PROCESSING

For a series consisting of numerical values, with a more formal saying for a distribution, there are some statistical properties that can be used for better defining and analyzing the given sequence. Some of these statistical properties can be listed as; average, variance, standard deviation, skewness and kurtosis. With the use of given properties, the target distribution can be defined with less parameters rather than telling all numbers in that distribution. This chapter basically aims to review the fundamental statistical methods used in the thesis study.

6.1 MEAN

In probability theory and statistics, the mean is the sum of the values divided by the amount of the variables. It basically shows the equal share of each member from the sum of this distribution. The basic equation for the mean is as follows;

$$\bar{x} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{6.1}$$

6.2 VARIANCE

In statistics the variance is the measure that is used for describing how far the numbers lie from the mean (the average). With another view it is a measure denoting

how far the numbers spread out in a given distribution. The basic equation for the "sample variance" is as follows;

$$S^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{(n-1)} \tag{6.2}$$

6.3 STANDARD DEVIATION

In statistics, standard deviation is a measure of the dispersion of a distribution from the average of this set. It is calculated from the variance equation, where the standard deviation is the square root of the variance. For a given distribution if the standard deviation is low, then it is the sign that the numbers in this set are close to the mean. The equation for the "sample standard deviation" is as follows;

$$S = \sqrt{S^2} \tag{6.3}$$

6.4 SKEWNESS

The skewness is the asymmetry measure of a given distribution for the horizontal axis. It can be negative, positive or zero. A negative skewness show that the distribution is right aligned where the positive skewness shows that the distribution is left aligned on the horizontal axis. The zero skewness states that the distribution is symmetric. The mathematical equation for the "sample skewness" can be given as follows;

$$\gamma_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^3}{(n-1).S^3} \tag{6.4}$$

6.5 KURTOSIS

The kurtosis is the vertical asymmetry measure of a given distribution, more clearly a kind of descriptor for the vertical shape of a probability distribution. A distribution is symmetric if the both sides look like the same according to the center point of that distribution. The kurtosis tells whether the data is peaked or flat relative to a normal distribution. A data set with high kurtosis shows that the given distribution has a distinct, vertical peak near the mean. Data sets with low kurtosis tend to have a flat top near the mean rather than a sharp peak. The mathematical equation for the "sample kurtosis" can be given as follows;

$$\gamma_2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{(n-1).S^4} \tag{6.5}$$

The kurtosis value for a standard normal distribution is 3. For that reason rather than using the normal kurtosis formula, more practically, the "excess kurtosis" formula is used by subtracting 3 from the kurtosis result as in follows;

$$\gamma_{2-ex} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^4}{(n-1).S^4} - 3 \tag{6.6}$$

With the use of excess kurtosis formula, a distribution is symmetric if the formula produces zero. A value greater than zero shows the distribution is peaked and similarly a negative value simply shows the distribution is flat compared to a normal distribution.

6.6 CHI-SQUARE STATISTICS

The Chi-Square statistics is one of the most used members of the nonparametric family of statistical tests which is mainly used for hypotheses testing. The *parametric statistics* test the hypothesis assuming the samples come from a normally distributed population where the *nonparametric statistics* test the hypotheses that do not require normal distribution as a main distinction.

The Chi-square statistics use nominal data. So instead of using means and variances, this test uses frequencies. Basically the Chi-Square statistics is used for two distinct circumstances such as; how an observed distribution differs from an expected distribution (mostly referred as goodness-of-fit test) or estimating the independency (difference) of two variables. The chi-square test always tests the null hypothesis, which states that there is no significant difference between the expected and observed results. Given the definitions, the basic computational equation for the Chi-square statistics can be stated as follows:

$$x^2 = \sum \frac{(Observed\ frequency - Expected\ frequency)^2}{Expected\ frequency} \tag{6.7}$$

CHAPTER 7

METHODOLOGY PROCESS AND EXPERIMENTS

This chapter aims to clarify the methodology of the pulmonary sound collection process, the technical details of the sound preprocessing work and the experiment steps that were used for diagnosing the asthma illness through predefined machine learning algorithms. Some of the techniques that were developed in the programming environment and the statistical methods that were used during each algorithm development phase are also explained within the corresponding titles.

7.1 THE SOUND REPOSITORY

The sound samples used in the study were recorded by the medical experts of Gaziantep University medical faculty under the supervision of Prof. Halil Rıdvan ÖZ, the head of the Genetics and Bioengineering Department of Fatih University. For the recording purposes a 2 channel DAQ card and two analog Sony condensed microphones were used to collect the pulmonary sounds from the chest region of the subjects. The two microphones were used to record the left and the right parts of the pulmonary region. Each record was sampled at 8Khz. as a single channel uncompressed (PCM) wave sound in 16bit depth. In the original experiment set, a population of 114 people was examined during a 2 years long study. Since all patients had different phenomena's, only 40 people were chosen from the input population with the medical proof that the chosen subjects had either Asthma illness or labeled as being in Healthy status. Thus 20 healthy and 20 asthma patient sounds were chosen from a population of over 600 sounds according to medical expert records.

7.2 PROCESS BLOCK DIAGRAM

During the study over the sound repository, for each experiment, four main process blocks have been applied to the whole repository sound samples to obtain the classification results. These four blocks can basically be named as; 1.filtering, 2.feature extraction, 3.feature selection and 4.experiments in RapidMiner environment. The main flow of the experiment process can be visualized as follows:

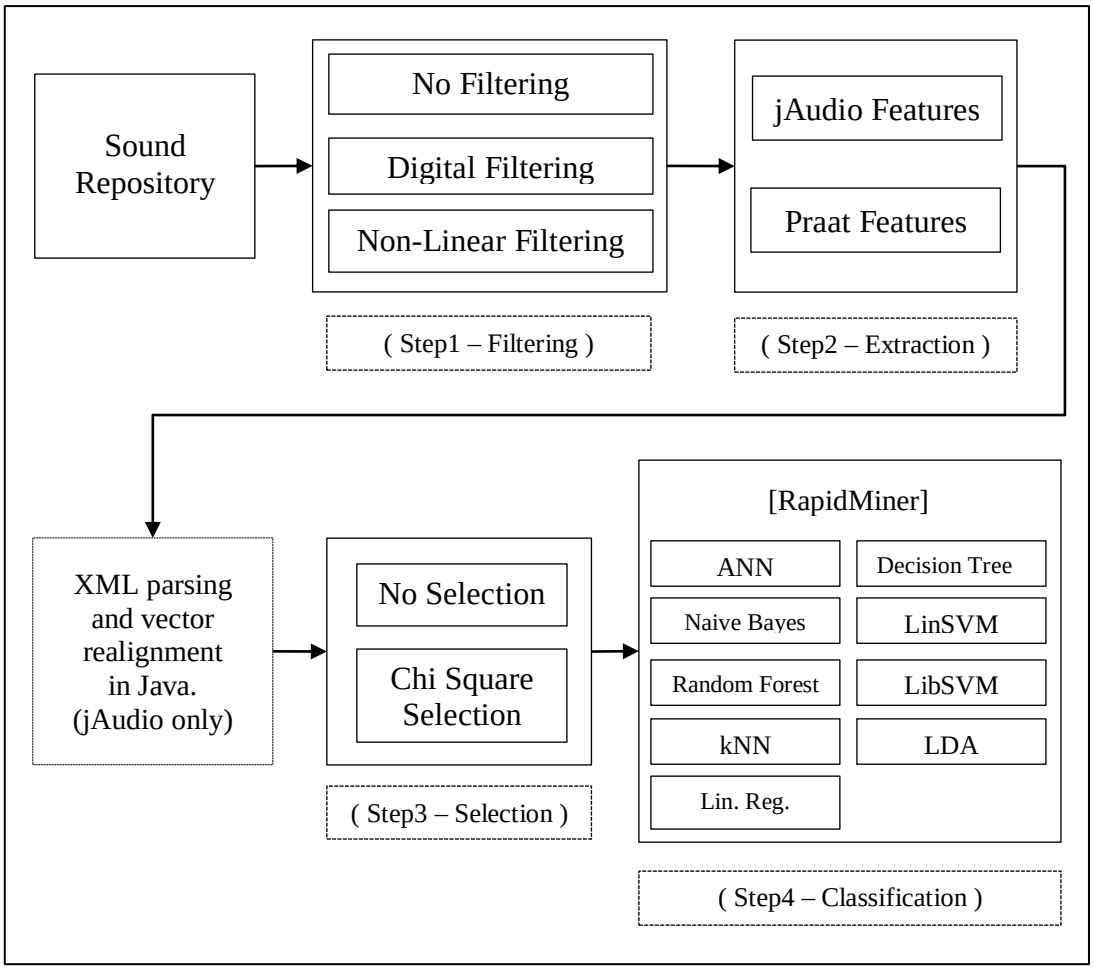


Figure 7.1 Experiment process block diagram.

As a note, in each process step, only one type of subcomponent is active at a time rather than a parallel processing. The output of the flow is an average accuracy table of nine classification algorithms.

7.3 STEP 1 – FILTERING

For the purpose of eliminating undesired time and frequency regions in the target signal and enhancing the classification accuracies, a "sound filtering" step was applied to the sound repository. This step mainly includes 3 types of filters with their sub types for each filter. The filter names used in this step can broke down as follows;

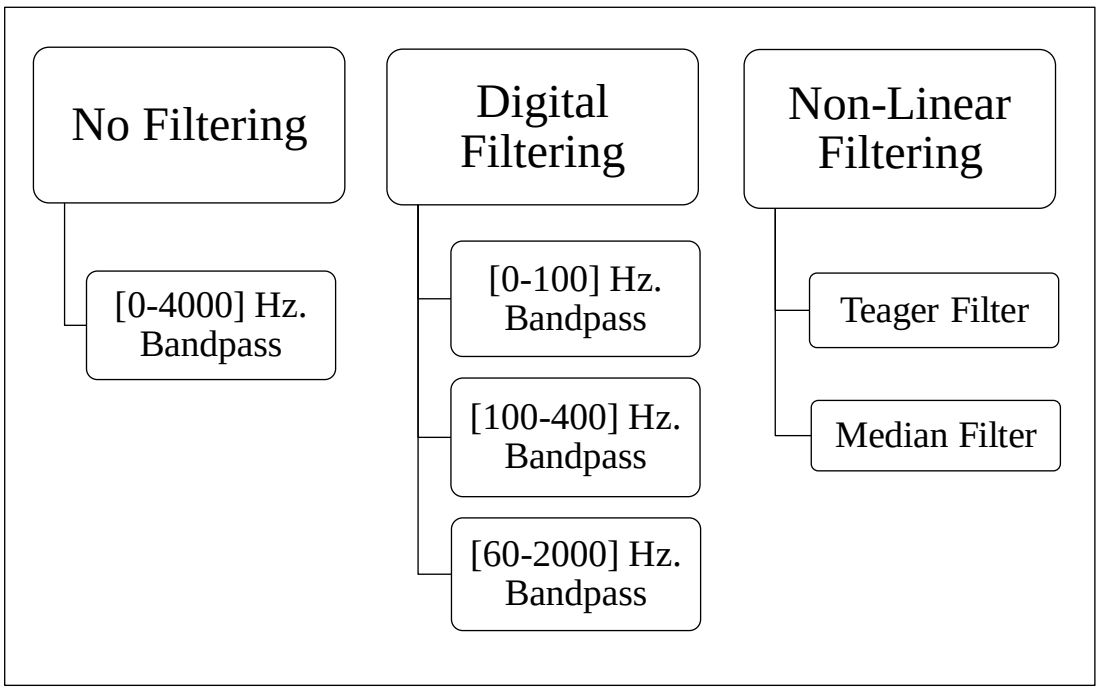


Figure 7.2 Data preprocessing block.

7.3.1 Teager Filter

The Teager filter is a homogenous, nonlinear, quadratic Volterra filter. This filter and its variations are mostly used in image enchantment applications such as contrast enhancing with good quality behavior. The basic mathematical equation for the Teager filter can be given as follows;

$$x[n] = x^2[n] - (x[n-1].x[n+1]) \tag{7.1}$$

7.3.2 Median Filter

The Median filter is used to remove noise in an image. This filter is mainly used over one and two Dimensional arrays. Since the sound files are one dimensional, only 1D algorithm was used within the thesis study. Median filters need a window length to operate. Since the length is user defined, the result is highly dependent to the user choice of the length, which is odd as a standard.

As the methodology, algorithm first parses the input signal into “predefined window length” small chunks. For each signal chunk, there is an overlap region consisting of neighbors from both sides. First the median of the chunk is determined. This median value is the target of the algorithm. Next the numbers within the chunk are sorted and the new median of the chunk is determined. As the last step, the old median is replaced with the new median value. Algorithm starts from the first member of the signal and continues until the last member. Thus all members of the signal are mapped from original domain into same size filter domain.

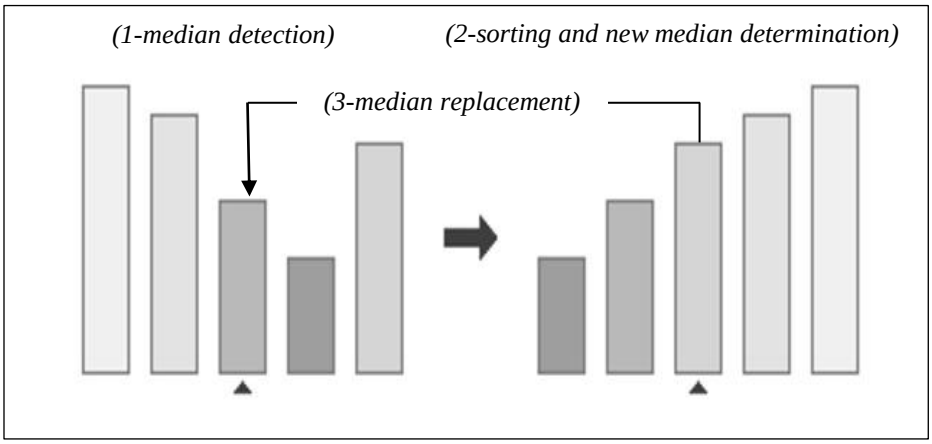


Figure 7.3 Median replacement process.

As a note, while the algorithm is running, there is a need for interpolation for the signal edges. This is mostly done by mirroring the border values to the negative directions. For the thesis study rather than implementing the Median filter algorithm, the “medfilt“ function of the Matlab Signal processing toolbox has been used.

7.4 STEP 2 – FEATURE EXTRACTION

As the most important step of the experiment methodology, the sound files were transformed into more descriptive vector forms with the use of feature extraction methods. For deriving the "MFCC", "LPC" and "Time domain" features, the jAudio program (jAudio, 2012) has been used. The jAudio program is an open source sound processing library implemented in Java language at McGill University. For the thesis study the windowing parameters of the jAudio program was chosen as 1024 samples for each window and 50% overlapping between the windows. As a standard, the output that is obtained from the jAudio program is an XML file consisting of multiple small window chunks. For each sound, there is a need for transforming these multiple vectors into a single feature vector. For achieving this goal first the MFCC output windows were aligned vertically under each other. Thus all same indexed members were ordered in same column. Then for each column; mean, plus standard deviation, plus skewness, plus kurtosis values were calculated as follows;

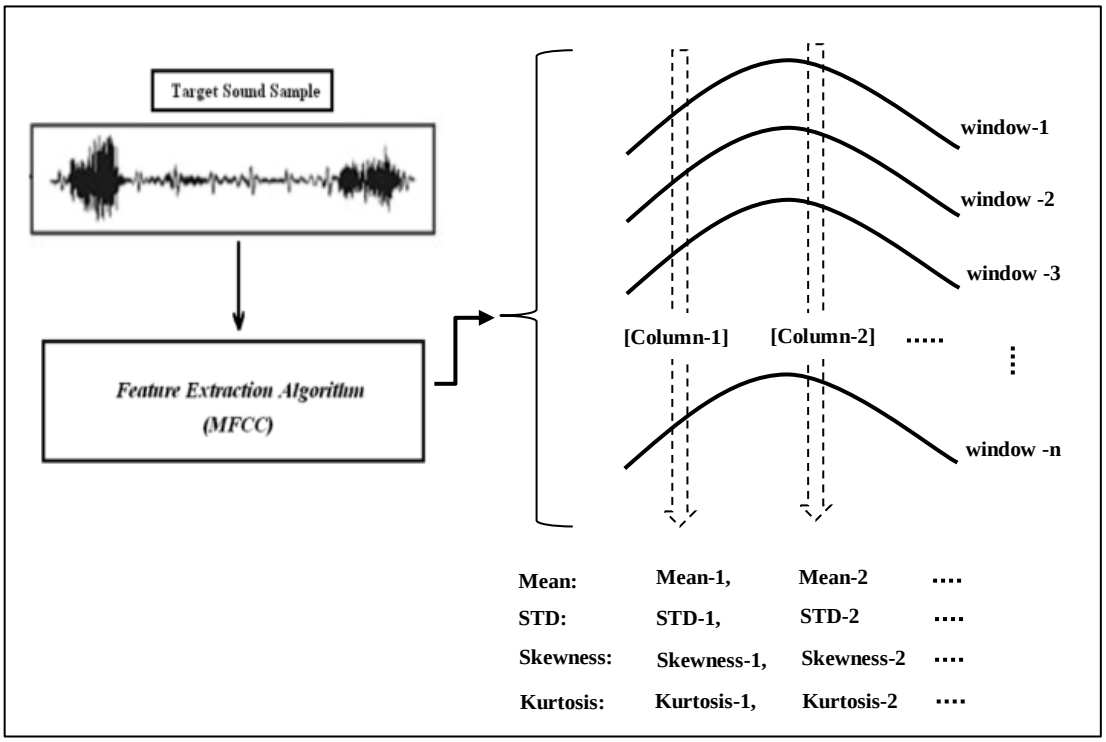


Figure 7.4 Feature extraction methodology.

While using the jAudio program, the XML output of the program was processed by a self-implemented program in Java environment (Java 7.u5 - Oracle, 2012) with NetBeans IDE (NetBeans 7.1, 2012). The program was designed object oriented and implemented in 7 packages and 24 classes. For all classes the code amount is more than 1000 lines in total. The general package and class layout of the program is as follows:

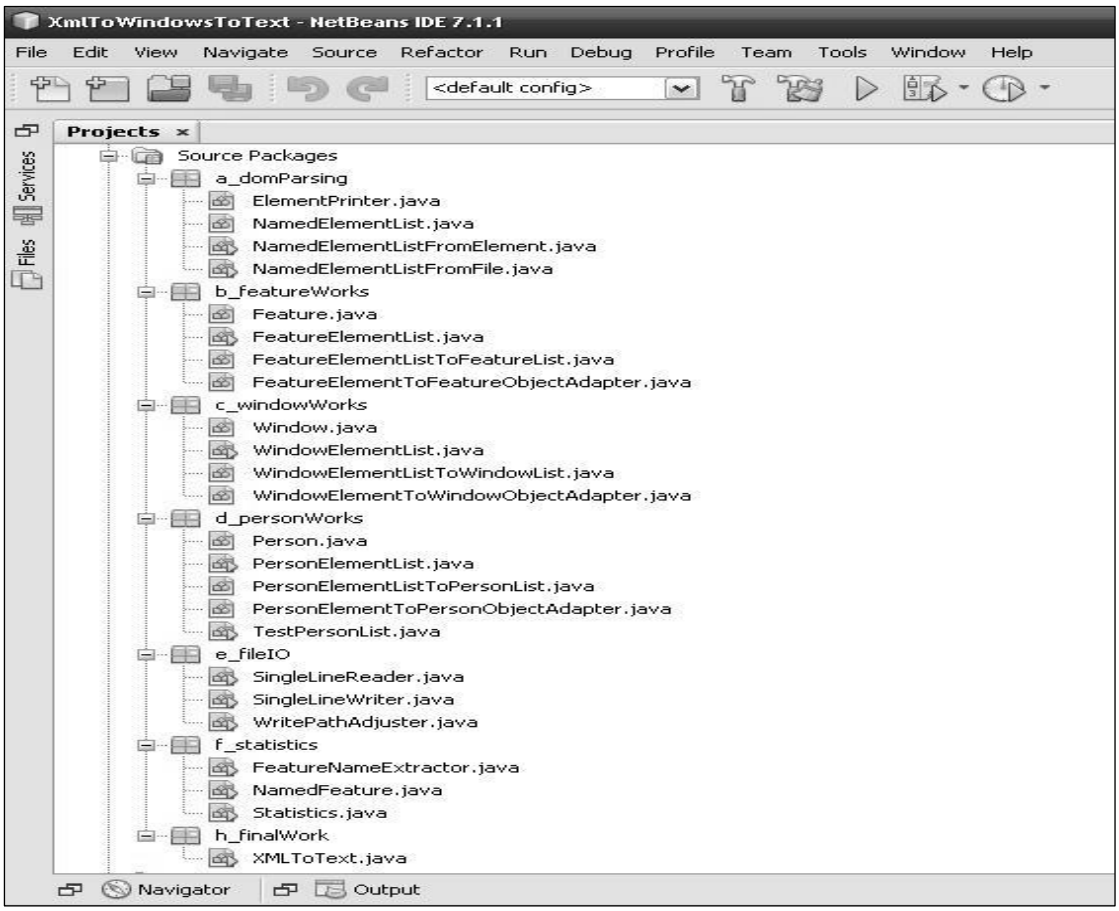


Figure 7.5 General layout of the parsing program in Java.

The Java program first parses the XML file with standard DOM API of w3c. Thus the file is read into the memory. Then the data inside the DOM nodes are mapped into the target objects. The output of the program is a text file consisting of one feature vector per sound. For the PRAAT program there is no need for future processing the input file since it includes only one feature vector for each sound sample.

7.5 STEP 3 – FEATURE SELECTION

After the feature extraction phase, each sound is now represented with a user defined length feature vector. For enhancing the classification results, a Chi-Square feature selection block has been used in RapidMiner environment between the feature extraction and the classification phases. The main idea behind using a selection block is removing possible uncorrelated features of a given vector.

For the thesis study, the sound repository was first introduced to the selection block and for all sound samples the same indexed features were then weighted by the Chi-Square selection block with a numeric value between zero and one. Upon the experiment, the top rated features were selected and fed to the classification block. The general layout of the Chi-Square block in RapidMiner environment is as follows:

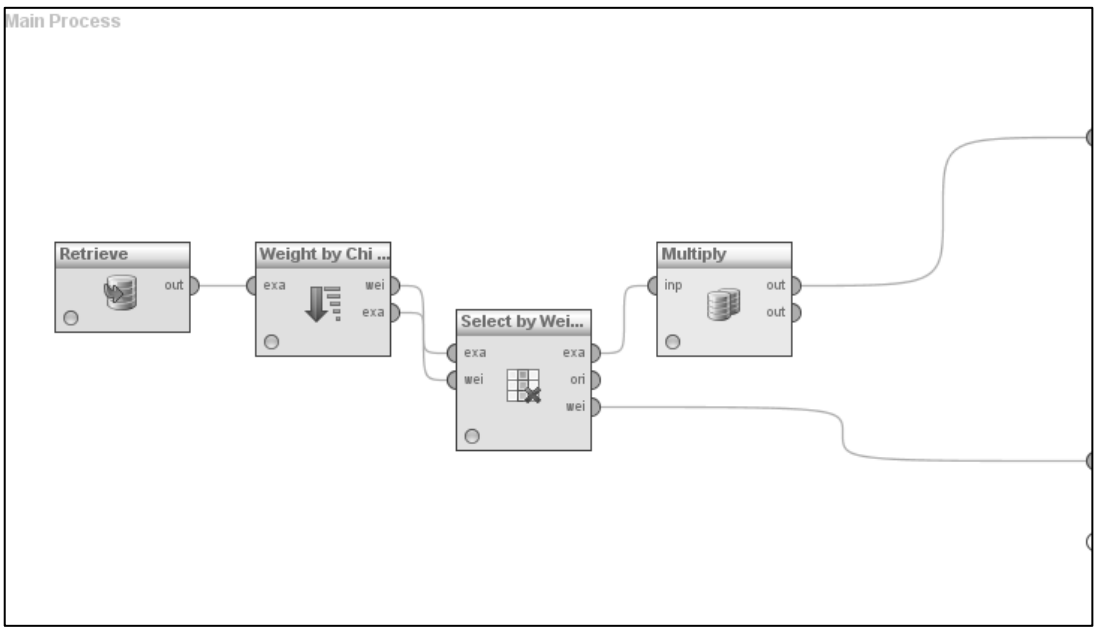


Figure 7.6 General layout of the selection block.

As a note, the feature selection block was used an optimization process after deriving the classification results without a selection block. For that reason the process block diagram also has a “no-selection” sub component in the corresponding block.

7.6 STEP 4 - CLASSIFICATION

The experiments were done in RapidMiner environment (RapidMiner 5.0, 2012). RapidMiner is an open source data analyses platform implemented in Java. It has a GUI interface for many most used machine learning algorithms and simulates an experiment environment easily. The general layout of the experiment environment in RapidMiner program is as follows;

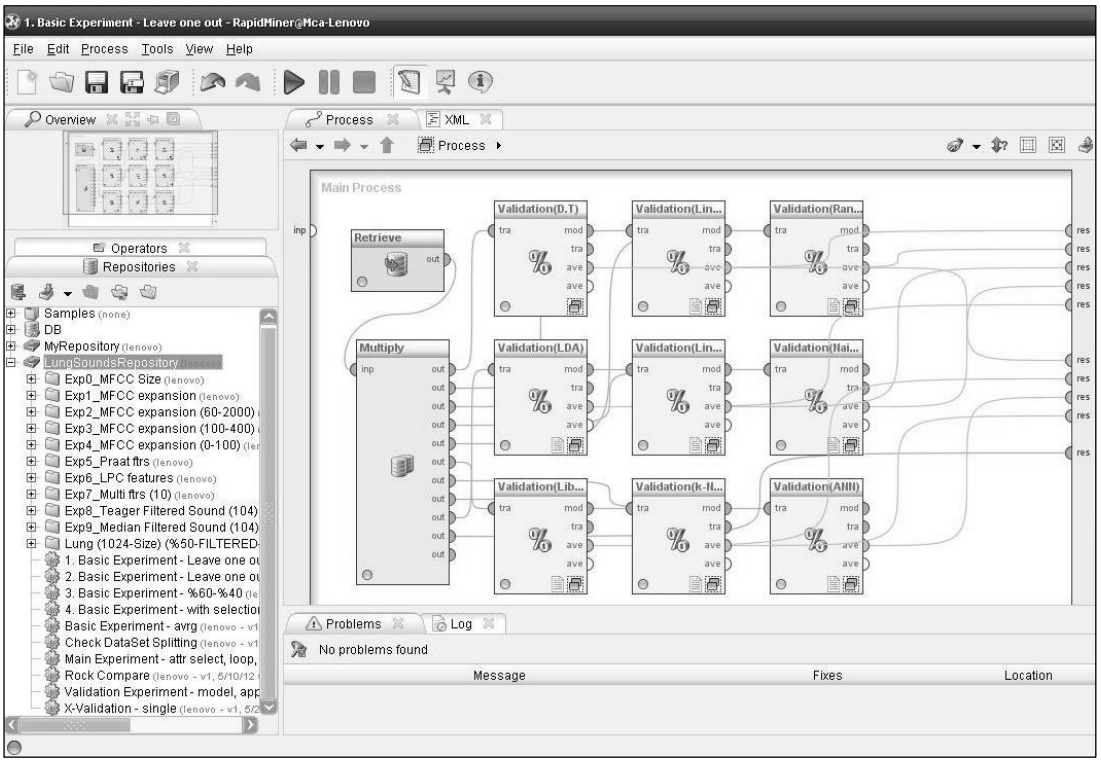


Figure 7.7 General layout of the experiment environment in RapidMiner.

As the first step in RapidMiner, nine cross validation blocks were constructed for the nine target machine learning algorithms. Rather than evaluating the each classifier's individual accuracy results, the average classification accuracy of the nine classification algorithms was evaluated for the thesis study. The classifier names that were used for the experiments are as follows:

Table 7.1 Nine classifiers used for the experiments.

No	Algorithm Name	RapidMiner Short Name
1	Artificial Neural Network	ANN
2	Naive Bayes	Bayes
3	Random Forest	RF
4	K Nearest Neighbors	kNN
5	Linear SVM	LinSVM
6	Support Vector Machine	LibSVM
7	Linear Regression	LinReg
8	Decision Tree	DT
9	Linear Discriminant Analyses	LDA

In the RapidMiner environment, each classifier validation block was first fed with the target repository sounds through a data multiplier block. These validation blocks were set to “Leave-One-Out” validation mode for the testing phase. Thus all classifiers were trained with the whole repository except one sample reserved for the testing. After the construction of the model from the training phase, the validation was done for the reserved sample and the process repeated for all members in the repository.

As an important note, each nine classifier produce a separate validation accuracy result. These nine individual results are evaluated as a single vote and the average of all votes is the final classification result.

7.7 EXPERIMENTS

The thesis study consists of 11 experiments for diagnosing the Asthma illness from the pulmonary sounds. In this title all experiments will be explained and the experiment results will be given. In each result graphics, the individual blocks give the average classification accuracy of nine classification algorithms.

7.7.1 Experiment 1 – MFCC Length Detection

Since the MFCC output length is the first parameter that should be found the aim of this experiment is the detection of the optimal MFCC coefficient length. For this reason five different MFCC vectors were derived from the unfiltered repository sounds and validated through leave-one-out and split (60% to 40%) validation blocks. The average accuracy result of the nine target classifiers is as follows;

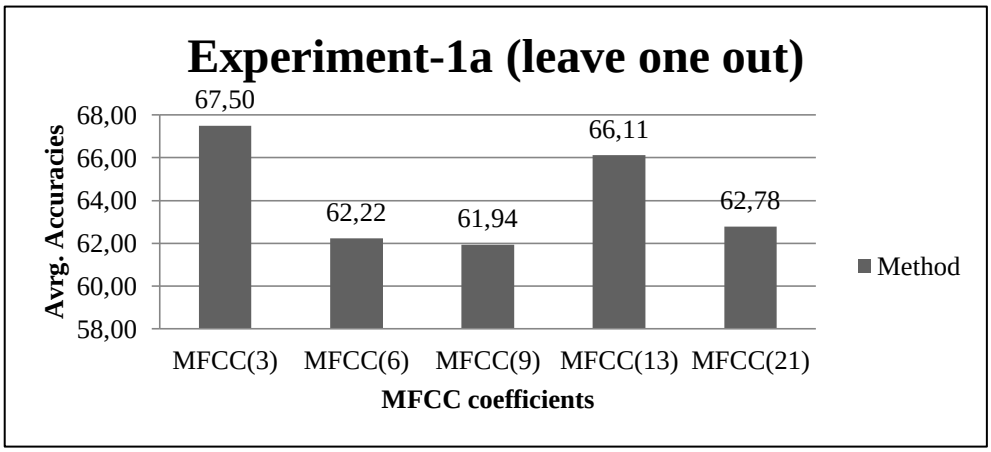


Figure 7.8 MFCC coefficient leave-one-out accuracy result.

According to the obtained results MFCC coefficients performs the best when the length is determined as 3. But since the leave-one out algorithm may not be the optimal case; the results were also queried with a (60% to 40%) split test as follows;

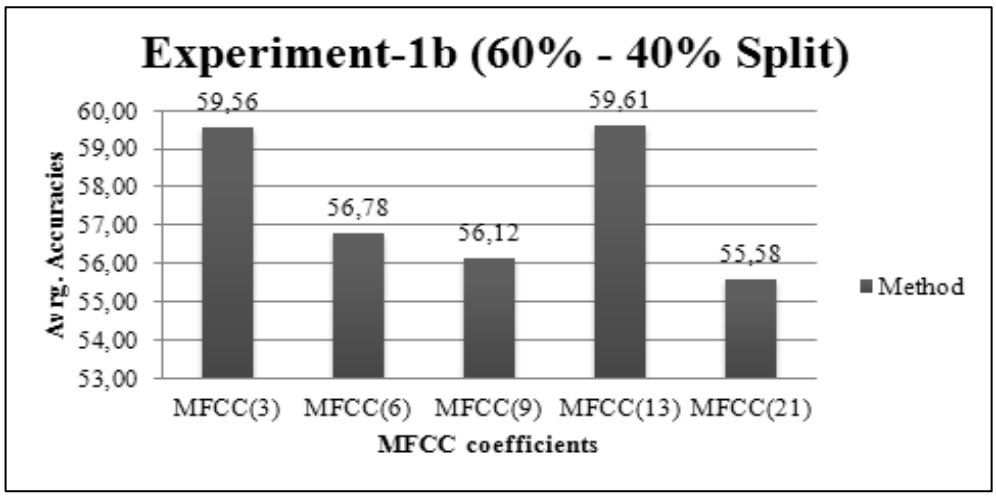


Figure 7.9 MFCC coefficient split validation accuracy result.

When the split validation is performed, the results broke for the favor of MFCC with length 13 and the average accuracy is lowered for the MFCC with 3 coefficients. By considering both cases, the final MFCC length was decided as 13 for the rest of the thesis study.

7.7.2 Experiment 2 – MFCC Features

The aim of this experiment is determining the optimal length MFCC feature vector for the target algorithms to produce the highest classification result. First, each unfiltered sound sample was presented to jAudio program to produce multiple MFCC or (MFCC with delta-MFCC) window chunks each having 13 columns. Then for each sound sample these window chunks were put under each other. Then for each window column; (only mean values) or (mean and standard deviation values) or (mean, standard deviation and skewness values) or (mean, standard deviation, skewness and kurtosis values) were calculated. Thus 8 different datasets were constructed. Lastly each vector was realigned and process repeated for the whole repository. For the nine classification algorithms, the average accuracy results are as follows;

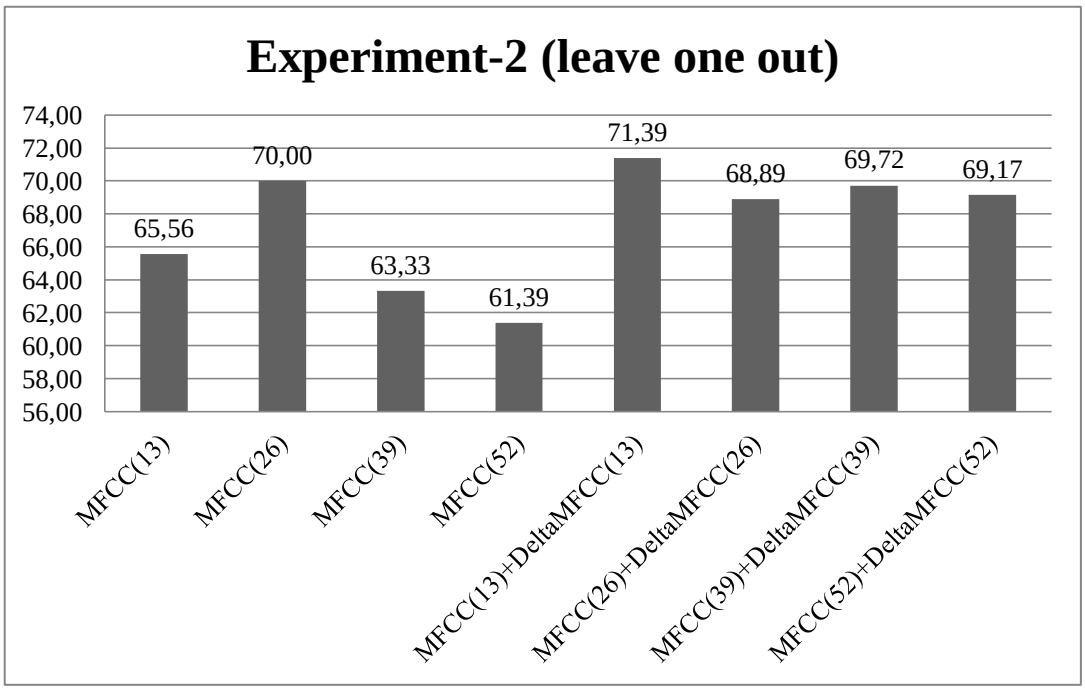


Figure 7.10 Optimal MFCC feature vector detection.

When the results are evaluated, the combination of MFCC and delta-MFCC features with 13 “column means” give the best average accuracy result of (71.39%) for the nine classification algorithms.

7.7.3 Experiment 3 – [60-2000] Hz. BandPass Filter Effect

The aim of this experiment is determining the best feature vector for the 60-2000 Hz. BandPass filtered sounds when the MFCC coefficients are 13. First the 40 repository sounds were filtered in the frequency domain with a [60-2000] Hz. digital filter. Then 8 different new datasets were constructed with the use of jAudio program. Each new dataset differs from each other with its vector type depending on the usage of column mean, standard deviation, skewness and kurtosis values. For the nine classification algorithms, the average accuracy results that were obtained from the RapidMiner environment are as follows;

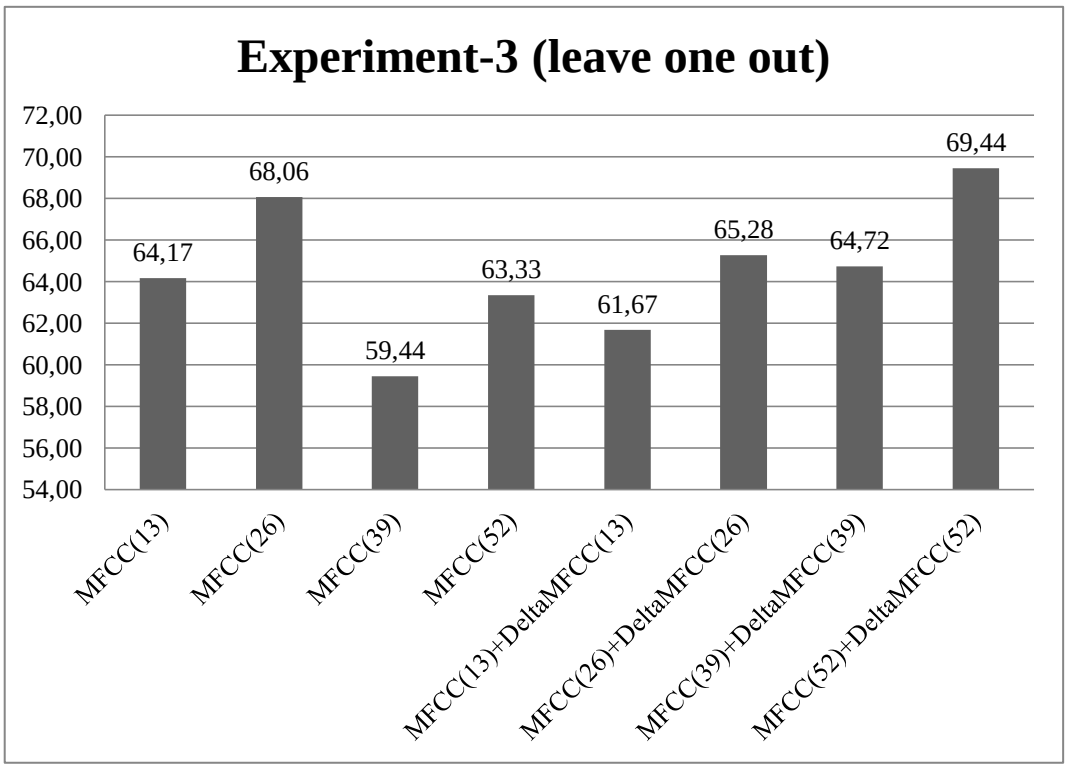


Figure 7.11 [60-2000] Hz. BandPass filter effect.

When the results are evaluated, for the nine classification algorithms, the combination of MFCC and delta-MFCC features with full column statistics (104 length feature vector) gives the best average accuracy result of (69.44%).

7.7.4 Experiment 4 – [100-400] Hz. BandPass Filter Effect

The aim of this experiment is determining the best feature vector for the 100-400 Hz. BandPass filtered sounds when the MFCC coefficients are 13. First the 40 repository sounds were filtered in the frequency domain with a [100-400] Hz. digital filter. Then 8 different new datasets were constructed with the use of jAudio program. Each new dataset differs from each other with its vector type depending on the usage of column mean, standard deviation, skewness and kurtosis values. For the nine classification algorithms, the average accuracy results that were obtained from the RapidMiner environment are as follows;

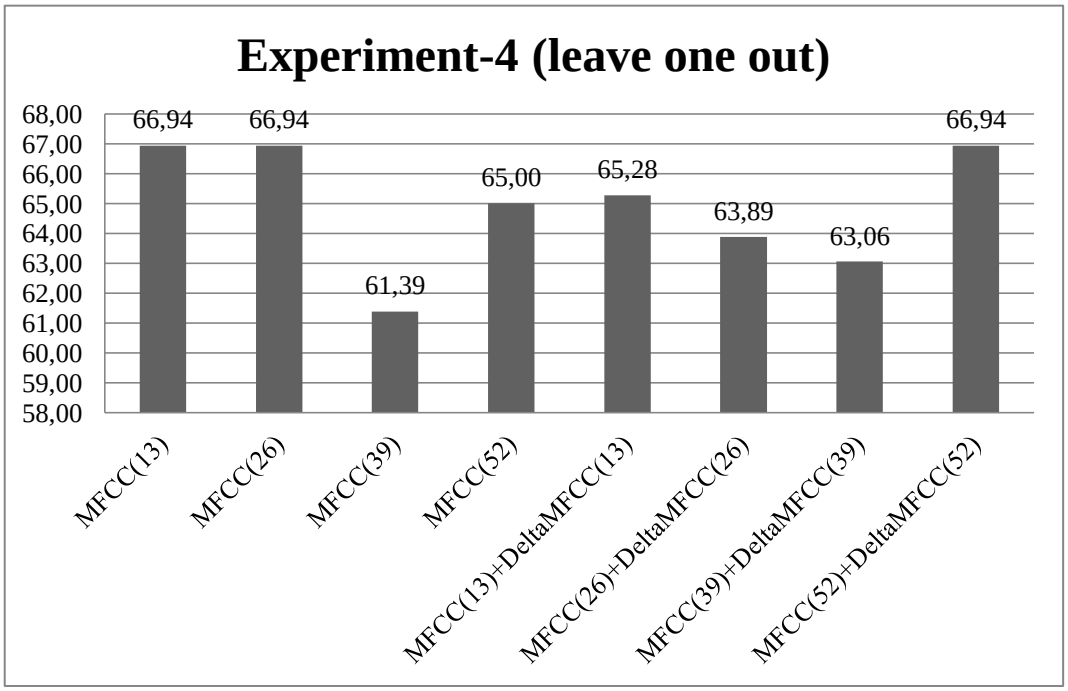


Figure 7.12 [100-400] Hz. BandPass filter effect.

When the results are evaluated, the combination of MFCC and delta-MFCC features with full column statistics (104 length feature vector) gives the best average accuracy result of (66.94%).

7.7.5 Experiment 5 – [0-100] Hz. BandPass Filter Effect

The aim of this experiment is determining the best feature vector for the 0-100 Hz. BandPass filtered sounds when the MFCC coefficients are 13. First the 40 repository sounds were filtered in the frequency domain with a [0-100] Hz. digital filter. Then 8 different new datasets were constructed with the use of jAudio program. Each new dataset differs from each other with its vector type depending on the usage of column mean, standard deviation, skewness and kurtosis values. For the nine classification algorithms, the average accuracy results that were obtained from the RapidMiner environment are as follows;

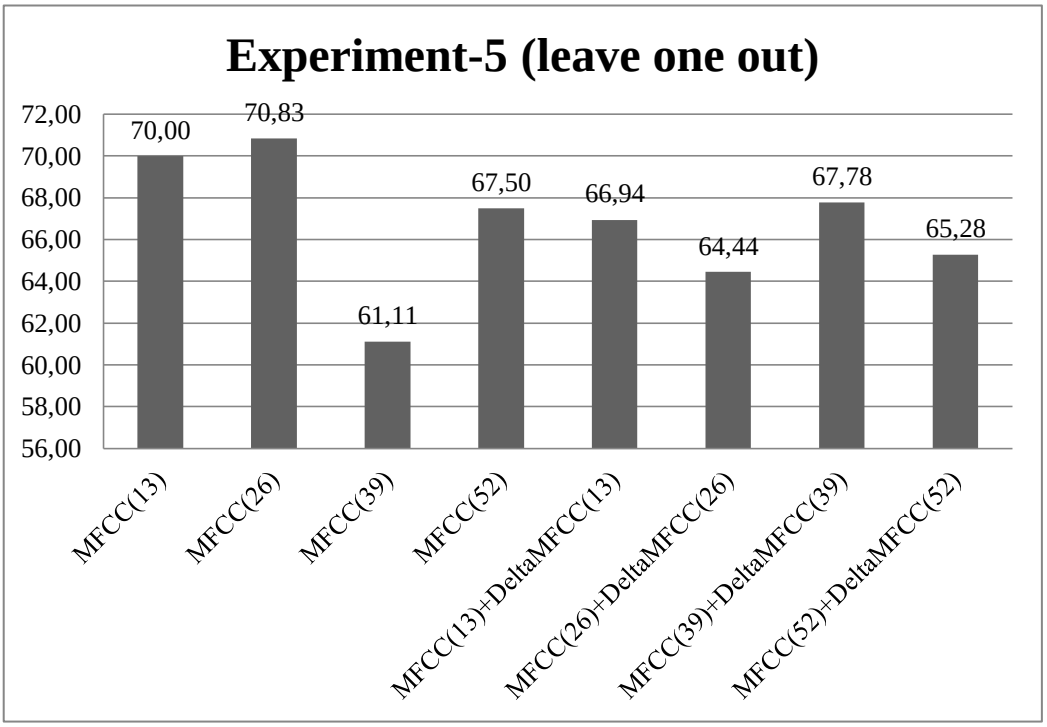


Figure 7.13 [0-100] Hz. BandPass filter effect.

When the results are evaluated, the MFCC features with 13 column means with 13 column standard deviation values (26 length feature vector) gives the best average accuracy result of (70.83%).

7.7.6 Experiment 6 – PRAAT Features

The aim of this experiment is to observe the average accuracy values when (17) PRAAT features are extracted from the "unfiltered" sounds. For this reason first the PRAAT features were extracted for each sound sample in the repository, and then feature vectors were transmitted into the RapidMiner environment. Lastly, with the use of standard experiment methodology, the feature set was introduced to the nine classification algorithms for the classification process. The average accuracy result that is obtained from the RapidMiner environment is as follows;

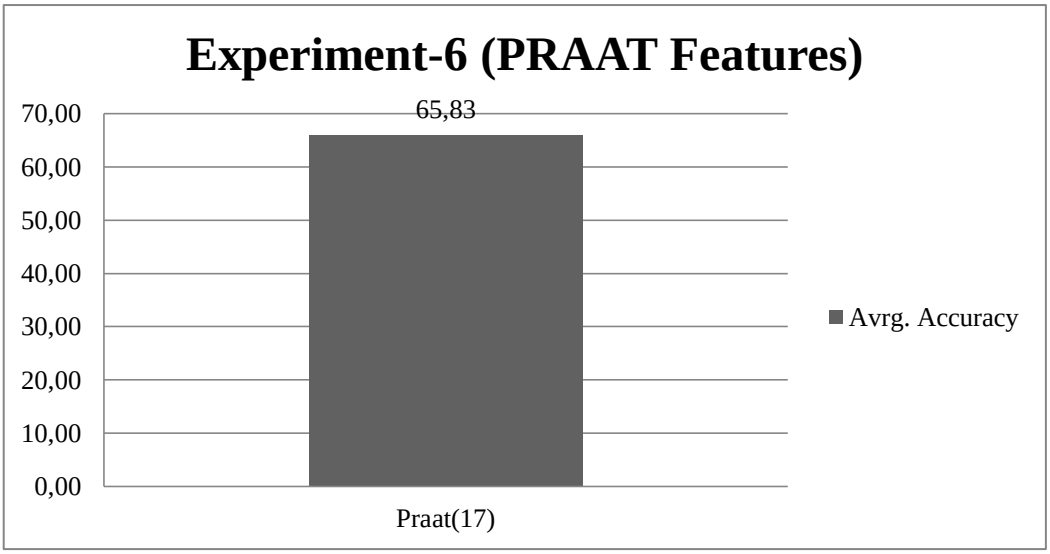


Figure 7.14 Average accuracy results for 17 - Praat features.

When the results are evaluated, the use of 17 LPC features gives an average accuracy result of (65.83%) for the nine classification algorithms.

7.7.7 Experiment 7 – LPC Features

The aim of this experiment is determining the best feature vector for the unfiltered sounds when the LPC coefficients are chosen as 13. For this reason LPC windows were step by step processed by statistical methods. The first derivative of the LPC windows (delta- LPC) was also taken into consideration while building the feature vector for each sound sample. In this case 8 different datasets were constructed. For the nine classification algorithms, the average accuracy results that were obtained from the RapidMiner environment are as follows;

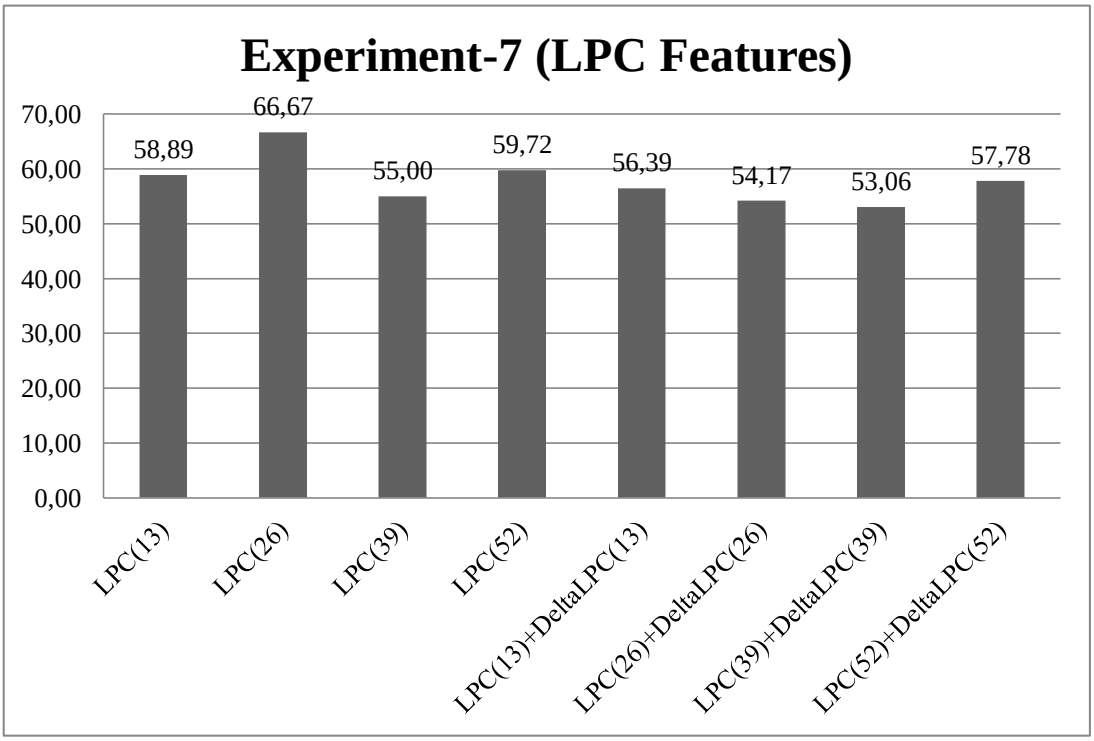


Figure 7.15 Average accuracy results for LPC features.

When the results are evaluated, the use of 17-LPC features with 13 column means and 13 column standard deviation values give the best average accuracy result of (66.67%) for the nine classification algorithms.

7.7.8 Experiment 8 – Time Domain Features

The aim of this experiment is observing the average accuracy values when 10 Time Domain features are extracted from the unfiltered sounds. For this reason after the extraction of the features with jAudio program, the feature vectors were transmitted into the RapidMiner environment. Then with the use of standard experiment methodology, the feature dataset was introduced to the classification process.

For the nine classification algorithms, the average accuracy results that were obtained from the RapidMiner environment are as follows;

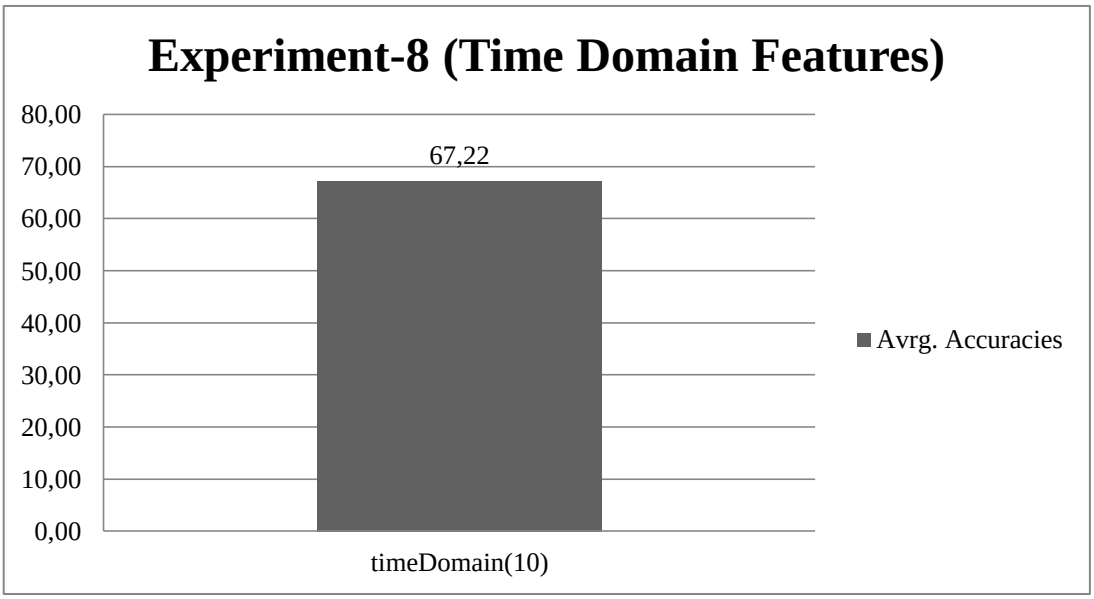


Figure 7.16 Average accuracy results for Time Domain features.

When the results are evaluated, the use of "10 time domain features" gives an average accuracy result of (67.22%) for the nine classification algorithms.

7.7.9 Experiment 9 – Nonlinear Filter Effect

The aim of this experiment is determining the best accuracy results for the nonlinear filter applied sounds when the feature extraction was done with 13 MFCC coefficients. For this reason first the repository sounds were filtered in Matlab environment with Teager and Median filters separately. Then the classifications were done for these new repository sounds individually.

The main idea behind this experiment is obtaining better classification accuracies by the use of nonlinear filtering techniques. For the nine classification algorithms, the average accuracy results that were obtained from the RapidMiner environment are as follows;

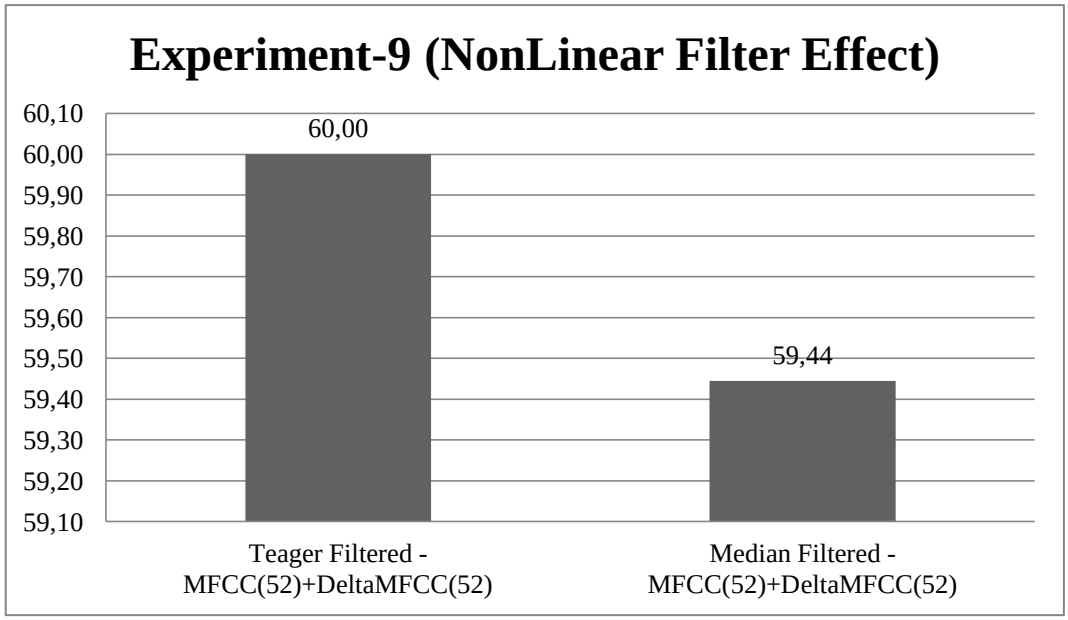
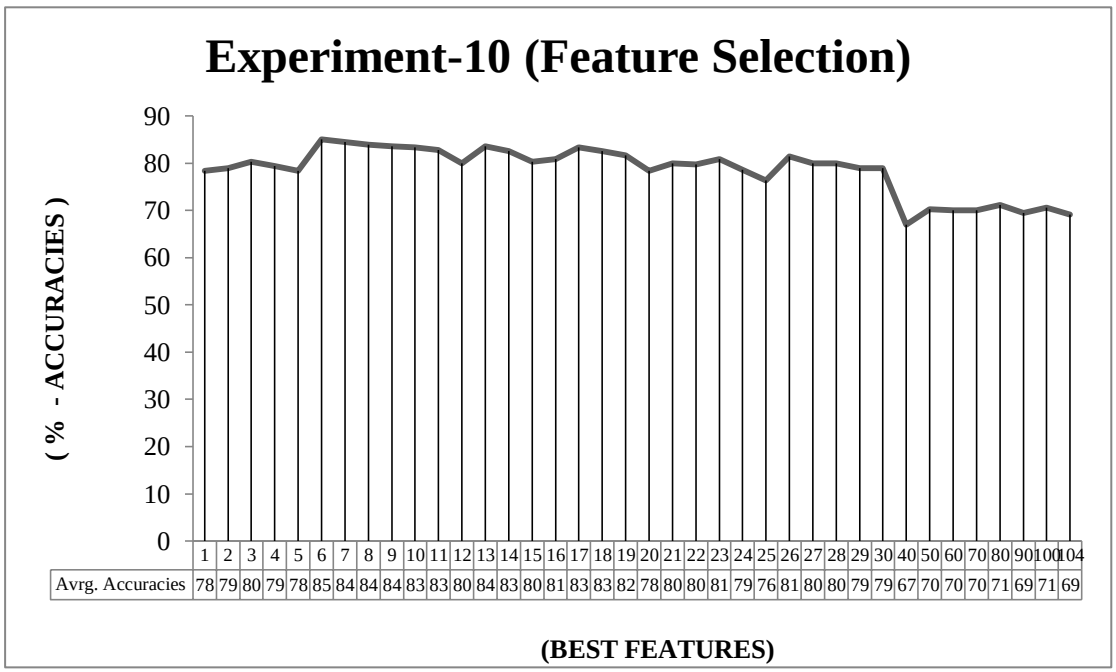


Figure 7.17 Average accuracy results for Teager and Median filtered sounds.

When the results are evaluated, the 104 length MFCC features derived from the Teager filtered sounds give an average accuracy result of (60%), where the 9^{th} order Median filtered sounds give an average accuracy result of (59.44%) for the nine machine learning algorithms.

7.7.10 Experiment 10 – Feature Selection with Chi Square Statistics

The aim of this experiment is eliminating the possible useless features from the target feature vector dataset for obtaining the best feature subgroup leading to a better average classification accuracy result. For this reason first the repository sounds were processed by the jAudio program and then the output of the program was parsed into 104 length feature vector dataset. Then in RapidMiner environment the Chi Square Statistic Feature Selection Block was used to rank the features with a grade from 0 to 1. According to the rankings, all 104 features were gradually experimented in such a way that; first the best single feature for all sounds was taken into account and the experiment done. Then in the next experiment one next feature was added to the feature vector. So step by step all feature combinations were tested and the results were kept. The average accuracy results that were obtained from the RapidMiner environment are as follows;



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	40	50	60	70	80	90	100	104
Avrg. Accuracies	78	79	80	79	78	85	84	84	84	83	83	80	84	83	80	81	83	83	82	78	80	80	81	79	76	81	80	80	79	79	67	70	70	70	71	69	71	69

Figure 7.18 Average accuracy results for feature selection process.

The main idea of this experiment is to obtain the best feature subgroup and its length from the 104 lengths MFCC plus delta-MFCC feature vector. The results show that for the best six features, the average accuracy is the highest around (85.0%).

7.7.11 Experiment 11 – Feature Selection for All Methods

From the former experiment it was obtained that the use of Chi Square Statistics method had increased the average classification accuracy for the unfiltered sounds and also reduced the feature vector length from 104 into 6. So the best average accuracy was obtained as (85.0%) for the nine machine learning algorithms.

In this experiment, the effect of feature selection process over digitally filtered sounds and over the non-MFCC feature extraction methods was queried with the same idea. So this experiment is two folds such that; in the first step BandPass filtered sounds were subjected to the MFCC algorithm and then the resulting feature vectors were simplified with Chi Square selection block for their best 6 features. And in the second phase, Time Domain features, PRAAT features, LPC features, Teager filtered sounds and Median filter applied sounds were also represented to the feature selection algorithm for their 6 best features.

In the first phase the average accuracy results that were obtained from the RapidMiner environment are as follows;

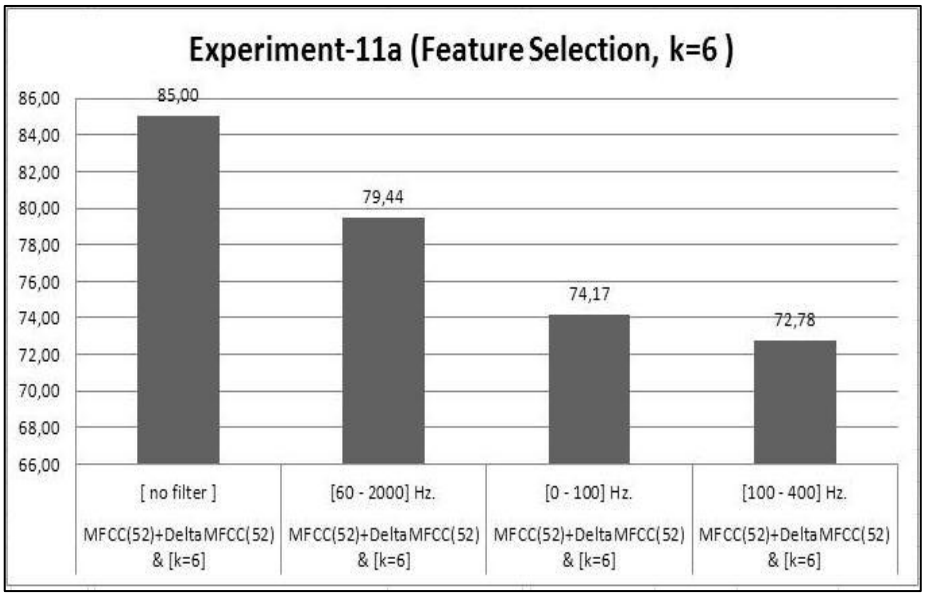


Figure 7.19 BandPass filter and feature selection results.

In the second phase, for 9 target classifiers, the average accuracy results that were obtained from the RapidMiner environment are as follows;

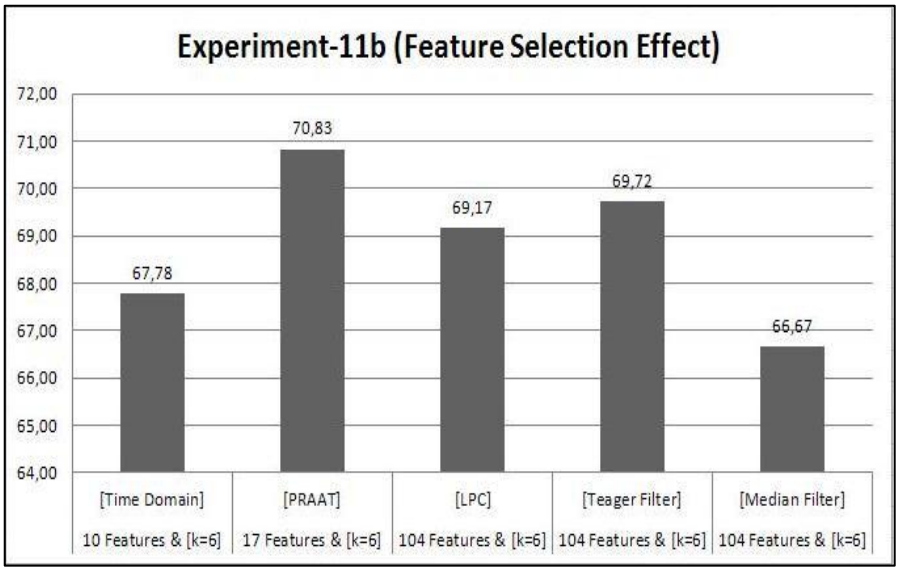


Figure 7.20 Non MFCC methods and feature selection results.

CHAPTER 8

RESULTS AND CONCLUSION

8.1 RESULTS AND DISCUSSION

During the thesis study 11 main experiments were done over the target lung sounds with thousands of sub observations. As stated before, the original sound repository had a large amount of noise which occurred during the sound capture process. In such a case the best classification accuracy was tested by using nine machine learning algorithms.

As a pre-process to the work, a noise removal process could be fine but that could also lead to some unwanted gaps in the original sounds. For that reason the best machine learning and feature extraction method should have the ability to handle these unwanted environment effects, such as noises and hysteresis, by eliminating the needs for future noise filtering techniques. The experiments consisting of multiple sub-observations, without any noise filtering process, have proven that the MFCC feature extraction algorithm achieved an acceptable average result. The proposed solution with 13 MFCC feature length with no filtering process seem to work efficiently for all possible learning methods. The jAudio program's behavior for evaluating the MFCC algorithm output window is summarizing the MFCC and delta-MFCC window columns with only column means and standard deviation values. In the study it was figured out that, adding column skewness and kurtosis values and thus expanding the feature vector to a more descriptive form helps the classifiers better evaluate the cases. From this well descriptive form it is also found out that, feature selection with Chi Square statistics method increased the average classification accuracies more than 10 percent.

8.2 CONCLUSION

In this study various machine learning algorithms were used for the diagnosis of Asthma disease from the lung sounds. The sound dataset was filtered with both digital and nonlinear filters for the purpose of eliminating the out of interest frequency regions and the effects of used filters were compared. For the feature extraction phase, four different extraction algorithms were evaluated and their performances on the classification algorithms were observed. While unifying the extraction algorithm outputs, various statistical methods were used. As an optimization phase Chi-Square statistical feature selection method was also applied onto feature vectors to receive higher accuracy values. The foundations of this thesis study experimentally prove that the proposed solution with the use of MFCC feature extraction algorithm with Chi-Square feature selection method promises an average accuracy value of 85% over all most used machine learning algorithms where the proposed solution can easily be implemented not only in computer environment but also in a future on chip device.

8.3 FUTURE WORK AND SUGGESTIONS

As the future study, the work can be expanded as to include more sounds in the repository. Having only 40 sounds may not include all possible Asthma patterns and this may also lead to a weak reasoning for the classification algorithms since the training was done in supervised learning paradigm. As a second point, the sound capturing process can be improved in such a way that the sound noise level is kept as low as possible. If the noise level could be lowered during the recording phase, the machine learning algorithms can better evaluate the asthma patterns rather than dealing with the comparison of high pitched noise levels. What is more, some sound matching algorithms aiming to keep the sounds perceptually balanced is planned to use for the same work to obtain better classification results.

REFERENCES

Ai, O. C., Hariharan, M., Yaacob, S., & Chee, L. S. (2012). Classification of speech dysfluencies with MFCC and LPCC features. Expert Systems with Applications, 39, 2157-2165.

Alpaydin, E. (2009). *Introduction to Machine Learning* (2. ed.). The MIT Press.

American Thoracic Society. (2012). Retrieved April 2012, from http:// www.thoracic.org/education/pulmonary-function-testing/index.php

Aydore, S., Sen, I., Kahya, Y. P., & Mihcak, M. K. (2009). Classification of Respiratory Signals by Linear Analysis. *31st Annual International Conference of the IEEE EMBS*, (s. 2617-2620). Minneapolis, Minnesota, USA.

Bahoura, M. (2009). Pattern recognition methods applied to respiratory sounds classification into normal and wheeze classes. *Computers in Biology and Medicine, 39*, 824-843.

Boersma, P., & Weenink, D. (2001). PRAAT, a system for doing phonetics by computer. *Glot International, 5*(9/10), 341-345.

Cohen, A., & Landsberg, D. (1984). Analysis and Automatic Classification of Breath Sounds. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, 31*(9), 585-590.

Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation, 19*(90), 297-301.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.

Cristianini, N., & Taylor, J. S. (2000). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods (1. ed.). Cambridge University Press.

Duch, W., & Jankowski, N. (1999). Survey of neural transfer functions. *Neural Computing Surveys*(2), 163-213.

Dunham, M. H. (2002). *Data Mining: Introductory and Advanced Topics* (1. ed.). Prentice Hall.

Feng, J., & Satttar, F. (2007). A New Automated Approach for Identification of Respiratory Sounds. *Multimedia and Expo, 2007 IEEE International Conference*, (pp. 356-359).

Fletcher, T. (2009). Retrieved 2012, from Support Vector Machines Explained: http://www.tristanfletcher.co.uk/SVM%20Explained.pdf

Golding, B. (1840). Advantages presented by the employment of a stethoscope with a flexible tube. *London Medical Gazette, 1*, 440-442.

Hassoun, M. H. (1995). *Fundamentals of Artificial Neural Networks* (1. ed.). The MIT Press.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation* (2. ed.). Prentice Hall.

Indiana University. (2012). *Danielson-Lanczos Algorithm*. Retrieved 2012, from http://hep.physics.indiana.edu/~hgevans/p411-p610/material/08_pde_iv/fourier.html

jAudio. (2012, July). 2.0. Retrieved from http://jmir.sourceforge.net/jAudio.html

Java 7.u5 - Oracle. (2012, July). Retrieved from http://www.oracle.com/index.html

Kröse, B., & Smagt, P. v. (1996). *An Introduction to Neural Networks.* University of Amsterdam.

Laennec, R. H. (1819). De l'Auscultation Médiate ou Traité du Diagnostic des Maladies des Poumons et du Coeur. Paris: Brosson & Chaudé.

Littmann Inc. 3M. (2012). Retrieved April 2012, from http://solutions.3m.com/wps/portal/3M/en_US/3M-Littmann/stethoscope/littmann-learning-institute/about-stethoscopes/stethoscope-history/

Lyons, R. G. (2011). *Understanding Digital Signal Processing* (3. ed.). Prentice Hall.

Martinez-Hernandez, H. G., Aljama-Corrales, C., Gonzalez-Camarena, R., Charleston-Villalobos, V., & Chi-Lem, G. (2006). Computerized Classification of Normal and Abnormal Lung Sounds by Multivariate Linear Autoregressive Model. *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference*, (pp. 5999-6002). Shanghai,China.

McKay, C., & McEnnis, D. (2012, July). 2.0. Retrieved from http://jmir.sourceforge.net/jAudio.html

Meng-Lun, H., Jen-Chien, C., & Feng-Chia , C. (2006). Respiratory Wheeze Detection System. *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference*, (pp. 7553-7559).

Minsky, M., & Papert, S. (1969). *Perceptrons: An Introduction to Computational Geometry* (1. ed.). Cambridge MA.: The MIT Press.

Mitchel, T. M. (1997). *Machine Learning* (1. ed.). McGraw-Hill Science/ Engineering/ Math.

National Asthma Council Australia. (2011). Retrieved April 2012, from http://www.nationalasthma.org.au/handbook/diagnosis-in-adults/detection-and-diagnosis#Diagnosis

NetBeans 7.1. (2012, July). Retrieved from http://netbeans.org/

Praat. (2012, July). 5.3.19. Retrieved from http://www.fon.hum.uva.nl/praat

RapidMiner 5.0. (2012). Retrieved July 2012, from http://rapid-i.com/content/view/181/190/

Rogers, K. (2010). Approaches to Respiratory Evalution And Treatment. In *The Respiratory System* (p. 196). Rosen Educational Publishing.

Rogers, K. (2010). Methods of Investigation. In *The Respiratory System* (pp. 199-210). Rosen Educational Publishing.

Rosenblatt, F. (1962). Principles of neurodynamics; perceptrons and the theory of brain mechanisms. Washington: Spartan Books.

Sen, I., Saraclar, M., & Kahya, Y. P. (2010). Acoustic Mapping of the Lung Based on Source Localization of Adventitious Respiratory Sound Components. *32nd Annual International Conference of the IEEE EMBS*, (pp. 3670-3673). Buenos Aires, Argentina.

Shabtai-Musih, Y., Grotberg, J., & Gavriely, N. (1992). Spectral content of forced expiratory wheezes during air, He, and SF6 breathing in normal humans. *Journal Of Applied Physiology, 72*(2), 629-635.

Shuiping, W., Zhenming, T., & Shiqiang, L. (2011). Design and Implementation of an Audio Classification System. *Procedia Engineering, 15*, 4031-4035.

Smith , S. W. (1997). The Scientist & Engineer's Guide to Digital Signal Processing (1. ed.). California Technical Pub.

Sovijarvi, A., Vanderschoot, J., & Earis, J. E. (2000). Standardization of computerised respiratory sound analysis. *Eur. Respir. Rev., 10*(77), 585.

Taplidou, S. A., & Hadjileontiadis, L. J. (2010, July). Analysis of Wheezes Using Wavelet Higher Order Spectral Features. *Biomedical Engineering, IEEE Transactions, 57*(7), 1596-1610.

the Editors of Publications International Ltd. (2012). Retrieved April 2012, from http://health.howstuffworks.com/diseases-conditions/rare/12-deadly-diseases-cured-in-the-20th-century.htm

The Lung Association of Canada. (2012). Retrieved April 2012, from http://www.lung.ca/diseases-maladies/help-aide/spirometry-spirometrie/index_e.php

The Merck Manual. (2012). Retrieved from http://www.merckmanuals.com/professional/pulmonary_disorders.html

Widrow, B., & Hoff, M. E. (1960). Adaptive Switching Circuits. *IRE WESCON Convention Record*(4), 96-104.

World Health Organization. (2011). Retrieved April 2012, from http://www.who.int/mediacentre/factsheets/fs310/en/index.html