University of Oxford

Third year project

Detecting sleep apnoea via non-invasive methods

Authors:

George Cochrane

Tuan Anh Le

Sophie Louth

Sachin Mylavarapu

Supervisor:

Dr. Frank Payne

May 20, 2014

Abstract

The abstract text goes here.

Contents

Chapter 1

Introduction

The division of labour among the group members is outlined in F Division of labour.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Chapter 2

The Condition

Obstructive Sleep Apnoea (OSA), sometimes called hypersomnia sleep apnoea syndrome, occlusive sleep apnoea [94], sleep disordered breathing [54], hyperventilation syndrome and Pickwickian Syndrome, however the latter three are discouraged because they are also used to describe other disorders, is a sleep disorder characterised by repetitive blockages in the upper airway during sleep, but with inspirational effort. These blockages can be apnoeas, full closure, or hypopnoeas, partial closure, both cause a restriction in airflow, which can lead to reduction in oxygen saturation and frequent arousal in order to re-establish airflow [4].

The upper airways include the nasal cavity; oral cavity; and pharynx, the area behind the tongue (see Figure 2.1). Blockages mainly occur in the pharynx, this is usually held open by the pharyngeal dilator muscles, although these relax during sleep in a healthy subject they are still able to maintain airflow, however in an OSA sufferer they provide insufficient force to prevent collapse [30]. Collapse only occurs during inspiration and is due to negative pharyngeal pressure. During rapid eye movement (REM) sleep there is further relaxation of the pharyngeal dilator muscle which can lead to longer apnoea and hypopnoea events.

Many factors affect the likelihood a patient suffers with OSA, however one thing is consistent in almost all cases; the pharyngeal upper airway size is smaller than in normal patients and often more elliptical with the long axis directed anterior-posterior, rather than laterally as it is in non sufferers, which alters the pharyngeal dilator muscle orientation leading to a mechanical disadvantage, this can have a number of causes including fat deposits and facial bone structure [47]. Overweight and obese patients often have fat deposits lateral to the pharynx, not always substantial but its positioning creates or reinforces elliptical shape, pharyngeal orientation and reduction in size.

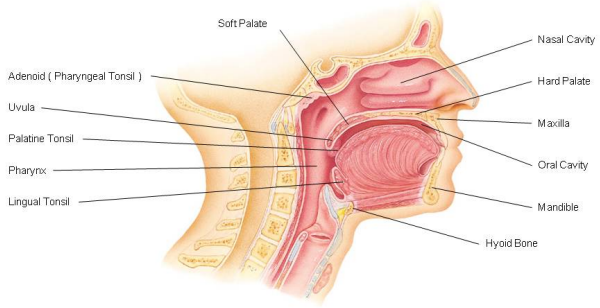


Figure 2.1: Sagittal View of the Face to show Upper Airways [49]

The main element of facial bone structure that influence upper airway size is the positioning and size of the maxilla and mandible (upper and lower jaw bones), these can influence the airway size in a couple of ways: micrognathia where the jaws are undersized, and retrognathia (or overbite) where the mandible is set back compared to the maxilla in both of these cases the tongue sits further back in the mouth, increasing tendency to block airflow. Lower positioning of the hyoid bone and Brachycephaly where the head is wider than it is tall can have a similar effect [**?**].

Abnormal facial tissues can also have an effect, large tonsils and adenoids, elongated or enlarged uvula (the dangly bit at the back of the mouth, see Figure 2.2), macroglossia (enlarged tongue), high arched or narrow hard palate, and reduced nasal patency (cross sectional area), possibly caused by nasal abnormalities, all have been shown to factor [82].

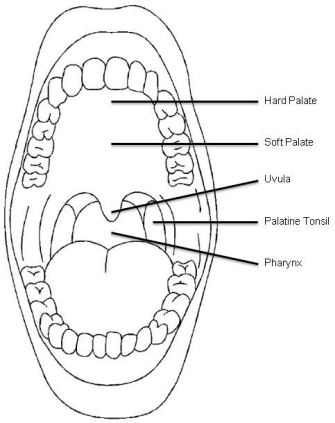


Figure 2.2: Anterior View of the Mouth to show Upper Airway Tissues [81]

Other factors include lying supine (on ones back) where gravity causes the tongue to fall into the airway. Some drugs including alcohol relax the pharyngeal dilator muscles more than just the effects of sleep, worsening OSA. Smoke irritates tissues including those in the upper airways, swelling them causing a narrowing of the airway [3].

Severity of the Condition can be measured in two ways AHI and RDI. AHI or Apnoea Hypopnoea Index counts the average number of apnoeas and hypopnoeas per hour asleep. The severity is then classified as minimal if AHI ¡5, mild if 5$\leq$ AHI ¡ 15, moderate 15

Chapter 3

Rationale

3.1 Medical Rational

Medical rational can essentially be split down into two areas: prevalence i.e. how many people suffer; and prognosis, likely course of the condition. There are a few other things that hint at the medical need for better diagnosis and others opinions of that.

The British Lung Foundation has prioritised OSA and created an OSA Charter calling on the UK governments to make OSA a national priority as well as encouraging investment in research. They held a three year campaign to raise awareness which ended in 2014 the campaign had two aims: to increase awareness both to the general public and health care professionals and to improve diagnosis. Part of the plan to improve diagnosis was to develop a national standard for diagnosis that would include a one stop shop so that the patient pathway would be reduced in length in order to reduce concerns about driving. A conference on OSA was held in February 2014 where doctors expressed desire to change the current system, due to increased need for doctor referral of patients to sleep centres [31].

PhysioNet and Computers in Cardiology with funding from Margret and H.A. Rey Laboratory for Nonlinear Dynamics in Medicine set up a competition with two prizes of $500 for whoever could classify ECG data, obtained minimally intrusively and inexpensively, into that from OSA sufferers and normal subjects, with the intension of using it as a screening tool [70].

The Agency for Healthcare Research and Quality felt that the diagnosis of sleep apnoea was a sufficiently important public health issue that they commissioned a study on future research needs which includes a reference to the need for portable monitors, including limited-channel, low-cost portable devices [7].

3.1.1 Prevalence

Prevalence of OSA is hard to know due to the fact a high levels of cases go undiagnosed (up to 90% of cases [26]), however estimates range from 1 to 28% depending on severity and location [104].

There may be an ethnicity element in prevalence, however few studies have been undertaken other than in western countries and therefore prevalence elsewhere is essentially unknown. For some areas it is known, this means studies on causes can be undertaken for example prevalence in Western Nations and Hong Kong is very similar but prevalence of risk factors is very different, there are high levels of obesity in the west but not in those studied from Hong Kong, so hypotheses have been produced on other risk factors including facial features being more prevalent in Hong Kong and some clinical observations support these hypotheses [40].

Gender has been shown to have an effect with estimates of 2 to 3 times greater risk for men compared to women, the reasons behind this are unclear [87]. Hormones have been considered but administration of the female hormones oestrogen and progesterone to men does not appear to have an effect [83]. Men show greater prevalence to many chronic diseases so this may be part of a greater trend and differences elsewhere have been shown to be linked to physical features, occupation, environment, attitude to health, and risky behaviour. There are gender differences in upper airway shape, muscle activity, facial shape, and deposition of fat in the airway, however the few studies that have looked at this have yet to find a conclusive link. Occupation, attitude to health and risky behaviour have not been studied in the context of gender disparity in OSA sufferers [93].

There are hypotheses proposing higher prevalence of OSA in pregnant mothers but few data sets to support this. Proposed mechanisms to cause this include excess weight gain and the effect of sleep deprivation on pharyngeal dilator muscle activity [32].

Although a positive trend between OSA prevalence and age appears to exist for mid life the same is not true for younger or older patients. OSA in children has similar consequence to that in adults and some of the pathophysiology (physical manifestation of a disease) is the same, however the etiology (causes) and associated morbidity (rate of incidence of a disease) can be very different, which means that it is generally studied independently from the adult form [2].

In old age prevalence of OSA increases however this does not necessarily mean that physiological changes associated with old age are causing OSA. If this was the case one would expect the prevalence rate to increase at the same rate as through middle age or at a higher rate. Figure 3.1 shows the Sleep

Heart Health Study on prevalence with age which starts to flatten in the 60s which suggests age related prevalence tails off at this point [103].

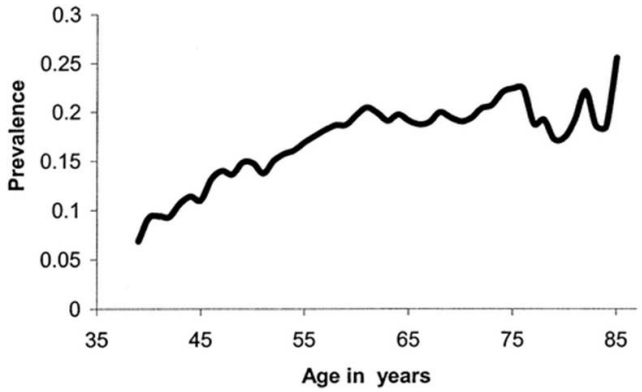


Figure 3.1: Prevalence of OSA by age in the Sleep Heart Health Study [105].

It is possible that older age OSA is actually distinct from that of middle age, several studies support this theory, as many of the key symptoms of middle age OSA are not present in the old age version including, daytime sleepiness, obesity, decrease in cognitive function and hypertension. Snoring is also significantly less reported however this could be caused by increase in bed partner hearing loss and death [22] [103].

3.1.2 Prognosis

The prognosis for sufferers of OSA can be quite severe with many suffering with secondary conditions such as hypertension, cognitive function is impaired and most suffer from daytime sleepiness which effects ability to drive and hold down a job.

There is an association between OSA and secondary hypertension (high blood pressure) independent of excess weight and other factors. This link is seen even in mild OSA, and given the prevalence of OSA could be having an impact on a significant proportion of those suffering with hypertension. However attempts to treat OSA in order to reduce hypertension have so far yielded unclear results [68].

Hypertension is linked to cardiovascular and cerebrovascular disease, given the link between OSA and hypertension, OSA will moderately contribute to the morbidity and mortality of these. There may also be direct links between OSA and cardiovascular disease however this has been less well studied. Whether treatment of OSA can improve cardiovascular disease has yet to be assessed [**?**].

Daytime sleepiness is a primary feature of OSA and many studies have shown that treatment of OSA does reduce daytime sleepiness [8]. Studies have found a significant association between snoring and

daytime sleepiness [21]. Snoring is a strong indicator of OSA, so the link between OSA and daytime sleepiness could be due to snoring, studies have shown the association between snoring and daytime sleepiness is independent of OSA [86].

The effects of OSA on cognitive function is not fully understood, there are some population based studies which find weaker correlations than clinic based studies, this is probably due to the biased population who attend sleep laboratories. In one study OSA was significantly but weakly related to reduced psychomotor efficiency (a measure of coordination of fine motor control with sustained attention), this link was not explained by daytime sleepiness [42]. In another study of self reported snorers a weak but significant association was found between OSA and neuropsychological function [1]. One suggested mechanism for reduction in cognitive function is due to oxygen starvation in the brain during apnoeas, this can change neurons especially in the hippocampus and right frontal cortex and so far has not seen improvement with treatment of OSA [34].

There is no specific quality of life measure for OSA although one is being developed, however the SF-36; a general health-related quality of life measure, a short form of the Medical Outcomes Study, is in use. A couple of studies have found a linear association between severity of OSA and decrements on the SF-36 scales, showing undiagnosed OSA has a similar affect on quality of life to other chronic disorders of similar severity, although another study showed a threshold effect as severity of OSA increased on a study of self-reported sleepiness or snoring, however a small sample size limits usability [27] [6].

Patients with OSA have a higher vehicle crash rate than the general population, this has been shown by crash records, self-reports and performance on driving simulators. This is a significant issue because it puts the lives of everyone not just the sufferers at risk. Studies undertaken in clinic will over estimate rate of crashes due to selection bias, however there are population studies looking at those with undiagnosed OSA which also show a strong correlation, especially among men. Self-reported sleepiness was not able to explain the crash rate. This is concerning because it means those at risk are not able to recognise it within themselves and are therefore unable to take precautions to reduce risk [25] [24].

3.2 Economic Rationale

3.2.1 Current Solutions

As is evident, sleep apnea, particularly obstructive sleep apnea (OSA) affects a significant proportion of the population and is worthy of attention. The current available diagnosis options for sufferers or

potential sufferers are limited, expensive, invasive and relatively inaccessible. We outline below the most common sources of help and diagnosis for potential sufferers:

- NHS - GP check-up, home devices and polysomnography (PSG)
- Commercial home-testing devices
- Other possible future options such as mobile sleep apps and non-contact health sensing technologies

NHS

According to the NHS guidelines, patients who believe they may be suffering from sleep apnea can schedule a visit to their GP. The GP will ask a few questions to determine the likelihood of apnea, along with a physical examination that includes a blood pressure (BP) test and a hypothyroidism test, to determine whether an underactive thyroid gland is the reason for the patient's tiredness. The patient can then choose if he/she would like to be observed for one night in a sleep centre, or would like to be given a monitoring device to wear at home when sleeping. Those who opt for a home sleep study are required to visit the sleep centre at a convenient time to collect and learn how to use the portable recording equipment. These include breathing sensors, heart rate monitors and oxygen sensors. Information from the device can be analysed on the next visit and further action, such as referring to a sleep centre can be taken [61, 62].

Observation at a sleep centre is done through polysomnography (PSG). This involves electrodes being placed on the face, scalp and above the lips, and bands being placed around the chest and abdomen. Additionally, sensors are placed on the legs and an oxygen sensor is attached to a finger. The tests carried out during a PSG include:

- Electro-encephalography (EEG)
- Electromyography (EMG)
- Recording of thoracoabdominal movements
- Recording of oronasal airflow
- Pulse Oximetry
- Electrocardiography (ECG)
- Sound and Video Recording

The data from these tests is used to positively diagnose obstructive sleep apnea and a treatment regimen is then decided upon and enforced by the healthcare professional [61]. For the purposes of this report, we shall not delve into the treatment of OSA - we concern ourselves with the diagnosis process and how we can improve it.

Commercial home-testing devices

While these devices are more commmon in the United States, they are also available in the UK, and can be purchased if desired without visiting a GP. An example of such home-testing devices is the AccuSom® test from NovaSom Inc. [63, 64], pictured below:

Figure 3.2: AccuSom® [64]

The prevalence of such tests in the U.S. points towards the increasing unfeasiblity of sleep centre tests for simple diagnosis of OSA, and is an indication of where the diagnosis process is headed in the future, in the U.K. as well as around the world. The economic reasons for the trend are clear. On average, a single night at a sleep centre and the associated tests could cost from $800 to $3000 in the U.S [41]. (figures for the NHS in the U.K. are more difficult to find, but should be comparable) The home tests can be administered at a fraction of that price, ranging from $200 to $600 [41]. Moreover, sleep centre tests are extremely inconvenient for the patients, and hence should be administered towards the later stages of the diagnosis process. The use of home-testing devices also allows a larger percentage of the population to be able to test for OSA, which is desirable given that around 5% suffer from undiagnosed OSA [62].

Other options

The trend towards greater user independence in OSA diagnosis is further observed through two major related breakthroughs - the rise of mobile sleep apps and the development of non-contact health sensing technologies.

The popularity of sleep monitoring apps in the iOS and Android App stores illustrates an increasing interest among smartphone users in their sleep patterns. Currently, such apps use the accelerometers and

microphones to detect movements and noises from the users while they sleep, and use relatively simple algorithms to determine which stage of sleep the user is in. In addition, some apps come with additional headsets or headbands that track electrical impulses and measure the user's activity more accurately. The alarm clock function is activated only when the user is in light sleep (within a reasonable time window) to ensure he/she is woken up feeling energised [65]. Some examples of such apps are:

- Sleep Cycle (iOS) - $1
- Sleep Bot Tracker (Android) - Free
- Wakemate - $60
- Lark - $99
- Zeo Sleep Manager Mobile - $99
- Sleep Tracker Elite - $149

The development of non-contact health sensing technologies holds promise as well. Recently, Xerox and Manipal University Hospital in India announced a collaboration to develop such techonologies at Xerox's research centre in Bangalore [88, 50]. Using image and signal processing algorithms, the collaboration aims to determine health indicators in a non-invasive manner, and such that monitoring can take place remotely [88]. The clear trend of health monitoring towards non-invasive methods and the increasing use of algorithms to supplement health care and diagnosis efforts is set to gain even more momentum as healthcare professionals begin to embrace 'big data' [69].

3.2.2 Proposed App - Where it fits in

The need for an app that is able to diagnose OSA from a mobile platform, using non-invasive techniques is clear from the current diagnosis solutions and trends. While there has been a shift towards cheaper home-testing devices before necessitating a visit to a sleep centre, the fact remains that even at $200, these tests are not inexpensive. What if we could create a means for diagnosing OSA without the need for cumbersome and invasive sensors, that could be accessed by anyone with a smartphone, at a negligible cost compared to a home-testing system? Those who have doubts over their sleep habits, but are too busy for a GP visit and do not want to spend money unnecessarily on a home-testing system, could try the app and move on to pursue the matter more seriously - if the results from the app suggested a need to do so. Since machine learning algorithms will be used in the processing of the data, the app has the potential to continuously improve in accuracy and, eventually, could outperform the home-testing devices. The development of image processing technology that could remove the need for invasive

monitoring, especially in neonatal care, by Manipal University Hospital and Xerox sets a precedent and raises the possibility of diagnosing OSA without the need for sensors being placed around the body.

3.2.3 Advantages of proposed app

The advantages of the proposed smartphone app, that will use machine learning algorithms to accurately diagnose OSA using non-invasive mobile phone sensors, are numerous. Firstly, such an app, if utilised as a precursor to home-testing systems and sleep centres, could result in significant cost savings for both the healthcare provider and the patient. From the point of view of the NHS, savings from performing polysomnograms only on those who really need it would be substantial. Moreover, some of the functions such as collecting information about the patient can be done through the app using questionnaires to save time for the doctors. From the point of view of the patients, not having to schedule visits to the GP for collecting and returning home-testing kits would result in a larger number of users for the app than would otherwise have been the case. This brings us to another advantage of using the mobile app - accessibility. Given that 65 % of people over 65 in the U.K. have OSA [62], along with a significant proportion of middle-aged men and women, and that the condition often goes undiagnosed, it is essential to reach out to as many users as possible. The best way to do so is through mobile phones. The number of smartphone users is projected to increase globally from 1.75 billion in 2014 to almost 2.5 billion by 2017 [19]. The graph below highlights the growth of mobile phone users worldwide:

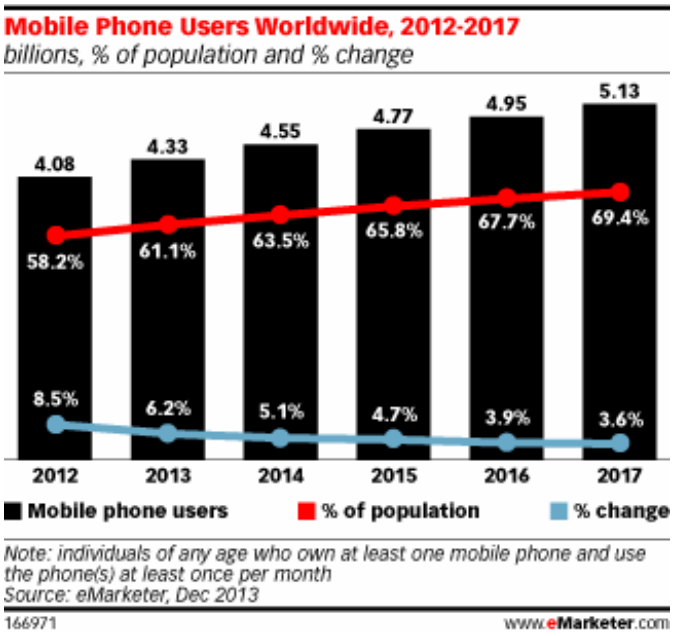


Figure 3.3: Mobile Phone Users 2012 - 2017 [19]

This highlights the potential for the app, and smartphone-aided diagnosis in general, not just in the U.K. but in other developing countries as well, where access to a sleep centre may not be as readily available.

The advantage that the proposed app holds over existing mobile phone apps is that it is specifically for diagnosing OSA - it is meant for medical purposes as opposed to general sleep cycle monitoring. This means that it does not compete with the above-mentioned apps, but serves a distinct purpose which is not possible in the other apps. Moreover, the use of machine learning algorithms in the app means that there is potential for the app to improve in accuracy over time as the amount and quality of training data used is improved. With every update, the app can become better at diagnosing OSA using only non-invasive methods.

In conclusion, the rationale for the project to develop such an app is clear. If developed, the app is viable as a supplementary service by the NHS and can improve the diagnosis of OSA in adults in the U.K. dramatically by reaching out to a wider audience and simultaneously result in savings for the NHS. It can also be further developed for use internationally, by those in developing countries in the future.

Chapter 4

The app

4.1 Structure Of The App

The app has been designed to have a very simple structure and layout so as to be immediately usable and comprehensible, even by those less technologically minded. Each activity branches from the main welcome screen activity, and upon completion of each activity, the user is the brought back to the welcome screen activity in order to select the next stage of the app to complete.

Figure 4.1: Network view of the app's structure

4.1.1 Home Activity

The app loads into the home activity when it is first turned on. The user is welcomed to the app and given a brief explanation of how to use the app. Below this the user is presented with three buttons to launch the three other activities.

4.1.2 Questionnaire Activity

The first task for the user to complete is the questionnaire, based on the eight STOPBANG criteria questions as detailed later in the Design Process section of this report. Pressing the submit button then takes you back to the start page and simply stores the number of yes answers the questionnaire received. This number is called during the processing of the overall results, noting that three or more yes answers indicates a high risk of Obstructive Sleep Apnea. The default for all of the answers is yes which should encourage the user to actively read/change each question (given that statistically the average person would have more no answers). It also means that should the user bypass the questionnaire section or ignore the questions before hitting submit, the app will use the higher risk criteria and results will be more likely to suggest too high a probability of apnea than miss a positive diagnosis altogether.

4.1.3 Sound Analysis Activity

This activity is indicated as the second task to complete, though it doesnt require completion of the questionnaire before being executable itself. The user is presented with four simple buttons as seen below. Note the following button restrictions:

- Stop cannot be pressed until the phone is recording audio.
- View Data and Delete File do nothing until there is data stored (Nothing yet is displayed as an indicator).
- When Start has been pressed, the Stop button becomes the only usable button. This is done by checking whether the inbuilt Boolean variable isRecording is true, and as such, the inverse becomes true again as soon as Stop is pressed.

For this simple prototype of the app, recording is restricted to one file in any instance of the app running. The sound data is automatically stored to a file accessible only by the app and is later analyzed by the machine-learning algorithm and overall results algorithm. Clearly the delete button is only included for cases where recordings were accidentally created and as such, it uses a comprehensive Are you sure? alert

upon pressing to avoid deletion of wanted files.

Progress bars are used whilst the sound is being saved and loaded which will reassure the user that the app is still running properly during these otherwise unresponsive stages of high CPU usage.

4.1.4 Results Activity

In this activity, a simple algorithm is used to determine an overall probability that the user has obstructive sleep apnea, and brief medical advice is given accordingly. It combines the results of the questionnaire and the audio recording analysis to do this, and therefore if one of these two parts is not present, it will ask for the user to go back and complete it before full results are displayed. Again, all analysis and loading tasks that takes more than half a second or so utilize a progress bar for improved user interface. subsectionNavigation Around The App The buttons are intended to be as intuitive as possible for navigation of the app. Loading each extra activity can only be done from the home activity, and the activity is exited by pressing the hardware back button whilst on the home activity. Each extra activity exits back to the home activity upon pressing this back button too, though the Questionnaire has the same functionality added in to the Submit button which helps reassure the user that the button has indeed been pressed.

Chapter 5

Design Process

5.1 Diagnostic Approaches

Before it is possible to work out how to design the app, the overall mechanism of diagnosis of the app needs to be decided as well as which approach to take in order to work out which signals and sensors to use for the app.

Diagnosis can be approached in a number of ways depending on whether the Doctor (or algorithm) thinks they recognise the condition or there is a standard flow chart for diagnosis, amongst other things. These approaches have been formalised and fall into four categories, they are; the algorithmic method, pattern recognition method, hypothetico-deductive method; sometimes called differential diagnosis, and the exhaustive method [66] [51].

The algorithmic method uses flowcharts and algorithms to analyse data that are precise and reproducible for example vitamin B12 level in blood. One follows steps making decisions at preselected branch point based on the clinical data available. This relies on a flow chart for the condition and for the condition to present in a relatively normal way. Abnormal presentation of a condition could easily lead to misdiagnosis. For example if an OSA sufferer is not overweight but instead has a high arched hard palate, they may well be missed by this method.

The pattern recognition method, is best for conditions with distinct presentation, especially those which present regularly in the population. It is an efficient method, especially useful when time critical diagnoses are needed. This does however risk jumping prematurely to a final hypothesis, although the consequences of this can be reduced by follow up in order to check the initial instincts.

The hypothetico-deductive method lends itself to primary care settings as it results in a differential

diagnosis, a rank ordered list of hypotheses. Hypotheses are generated and rejected as more data are collected and questions asked until a working hypothesis is reached. This method is most often used as it reflects how most people deal with life, and is therefore the most natural. It can cause errors if a doctor has been exposed to a certain diagnosis recently, biasing their importance weightings of certain symptoms. For example if a doctor has diagnosed a number of patients with OSA recently (or read a paper about it) they may diagnose a patient who expresses day time sleepiness and witnessed apnoeas with OSA when in fact the previous traumatic head injury would be an indicator for Central Sleep Apnoea, in this case a sleep test would be needed to see whether the patient was actually attempting respiration during apnoeas. Another error situation can be caused by the doctor failing to think about the probability of a diagnosis being correct, for example, if a patient presents with witnessed choking during sleep as the primary symptom rather than daytime sleepiness or witnessed apnoeas the doctor may diagnose Sleep Choking Syndrome however this has a much lower prevalence than OSA, it is considered rare by AASM [4].

The exhaustive method works on the premise of having all the information, all data collected, all questions asked. Completeness of data is important for hospitalised patients but acquiring it is too time consuming and expensive for most cases. It is useful for unusual conditions when other methods have failed or for unusual expressions of conditions.

The hypothetico-deductive method will be useful in order to establish what symptoms and signs can be used to differentiate OSA from other disorders. However the app itself will need to rely on pattern recognition, using standard and obscure symptoms of OSA to confirm diagnosis and rule out other options (or not). It would be unnecessary to use an exhaustive method as the app is only designed to pick up OSA not all sleep disorders or reasons for daytime tiredness, and this would be likely to frustrate a user.

5.2 Symptoms of OSA

Obstructive Sleep Apnoea sufferers display a wide range of symptoms and it is neither possible nor necessary to detect them all in order to diagnose the condition. This section will look at which symptoms to use and start to look at mechanisms for detecting them. Table 5.1 shows a list of all the presenting symptoms, secondary symptoms and exacerbating factors of OSA [23] [13] [4]. The secondary symptoms are things that the patient may have noticed but has not associated with their presenting symptoms, a patient may not experience all of these secondary symptoms, and they are not sufficient without at least one primary symptom to suggest OSA. However they are easy to assess using a questionnaire and are worthy of further work. The exacerbating factors are mainly physical features that a medical practitioner would need to assess; as such they would not be suitable to be assessed as part of this app with the exception of high BMI, large neck circumference, and potentially overbite as these could be assessed by a patient.

%width of table will need manual changing

Primary presenting symptom	Daytime sleepiness
	Loud snoring
	Witnessed apnoeas
	Witnessed arousals (choking/gasping)
Secondary symptoms	Irritability
	Dry mouth upon wakening
	Disorientation upon wakening
	Mental dullness upon wakening
	Grogginess upon wakening
	Incoordination upon wakening
	Nocturia (wetting the bed)
	Impaired cognitive function
	Morning headaches
	Personality change
Exacerbating factors	High BMI
	Large neck circumference
	High blood pressure
	Overbite
	Enlarged tongue
	Enlarged tonsils
	Elongated or enlarged uvula
	Nasal abnormalities
	Small jaw size
	High arched or narrow hard palate

Table 5.1: symptoms and exacerbating factors of OSA.

That leaves the primary presenting features these are the key features to use to distinguish OSA from other disorders. Table 5.2 shows disorders that also display one or more of these symptoms along with how they can be differentiated from OSA. From the table it is clear most of these conditions can be ruled out by diagnosis focussing on daytime sleepiness, snoring, and an apnoea awakening/choking combination [4].

Disorder Similar to OSA	Key OSA symptoms shared	Differentiator f
Narcolepsy	Excessive Sleepiness	Any night time
Idiopathic Hypersomnia	Excessive Sleepiness	Any night time
Insufficient Sleep Syndrome	Excessive Sleepiness	Any night time
Periodic Limb Movement Disorder	Excessive Sleepiness	Although parti
Depressive Episodes	Excessive Sleepiness	Any night time
Central Alveolar Hypoventilation Syndrome	Excessive Sleepiness, Choking, Snoring, Apnoeas	Hyperventilati
Central Sleep Apnoea	Excessive Sleepiness, Snoring, Apnoeas	Non prominent
Cheyne-Stokes	Choking	
Panic Attacks	Choking	
Sleep Choking Syndrome	Choking, Snoring	No apnoeas pr
Sleep Related Laryngospasm	Choking, Snoring	Strider present
Sleep Related Gastroesophageal Reflux	Choking	No apnoeas, m
Sleep Related Abnormal Swallowing Syndrome	Choking, Snoring	Gurgling rathe
Chronic Obstructive Pulmonary Disease	Snoring	May coexist w
Sleep Related Asthma	Snoring	Coughing (pro
Congenital Central Hypoventilation Syndrome	Snoring	Detected in ch
Sudden Infant Death Syndrome	Snoring	Detected in ch
Infant Sleep Apnoea	Snoring	Detected in ch
Altitude Insomnia	Snoring	The height (40

Table 5.2: Conditions which present at least one of the primary symptoms of OSA and differentiator from OSA.

Ascertaining these would eliminate all but central sleep apnoea and altitude insomnia however both of these would be picked up a significant proportion of the time by a questionnaire because the sufferers would only fit the risk cases for OSA in a small percentage of cases. Especially for altitude insomnia as those with a BMI over 35kgm^{-2} are unlikely to be climbing over 4000 m in height for extended periods (i.e. climbing mountains rather than flying).

Those suffering from central sleep apnoea are likely to reflect population distribution for BMI as weight is not a cause or exacerbating factor [4]. I.e. about 25% [20] of central sleep apnoea patients are obese (BMI ¿30) compared to 70% of OSA sufferers [48].

Table 5.3 suggests methods of detecting daytime sleepiness, snoring, apnoeas and arousal events, and distinguishes whether they are invasive or not [39]. The only viable approaches are video, accelerometry, audio and questionnaires, and so these will be investigated further next.

Symptom	Identifying Feature
Daytime Sleepiness	Patient self aware
Loud Snoring	Bed partner observed
	Rasping sound
	Airflow
Apnoeas	Oxygen Saturation
	Absence of sound
	Heart rate increase
	Electrical activity in the brain, change from low voltageduring REM to higher voltage during
	Jaw activity, relaxed during REM
	Eye movement, rapid during REM, decreased during arousal
Arousals	
	Sudden movement of arousal
	Choking / gasping

Table 5.3: Sensors for detecting primary symptoms of OSA.

5.2.1 Video

The aim is to pick up on apnoeas via cessation of breathing and arousal via significant body movement. Snoring would not be possible to detect although some other attributes that hint at OSA may be detectable e.g. lying in the supine position (on back). Video could also be used to detect sleep and activity, e.g. sleep cycle transitions and limb movements.

Merits include: smartphones already have cameras so an external camera would not be needed, although the cameras may struggle with the low light level in the sleeping space, and changing the light level in order to accommodate the camera could have a knock on effect on the quality or type of sleep.

Weaknesses include: positioning of the camera will be important and this is hard when all rooms are different and smartphones do not naturally stand up by themselves. Manual analysis of the video has been used until recently so there is not a great deal of statistical data on how well computer algorithms can detect OSA [77]. Video produces large data files that cannot be sensibly stored on a phone. Video has formed part of the polysomnogram in some areas for a long time as an aide to observation and only recently investigated as an independent method of diagnosis using automated analysis [12], as a consequence lots of video data does exist however it is highly regulated due to patient confidentiality, and potentially not labelled resulting in a need for a large number of specialist man hour to generate training and testing data sets [67].

5.2.2 Accelerometry

This has the potential to be able to detect body position, and arousal events. It has not been part of the polysomnogram although it has started to be investigated as a mechanism for detecting OSA. There is very little data available which makes a retrospective study a challenge, especially as there is no data combined with polysomnogram data for verification. Positioning of the phone in order to get the best results without damaging the phone would require some investigation. Non standardised arrangements of mattress and bed clothes may cause issues.

5.2.3 Audio

The most obvious way to detect audio signals from the patient is via the phones in built microphone although external microphones should also be investigated. It ought to be possible to detect apnoea arousal events and snoring.

Lots of work has been done on audio analysis of speech so there are lots of ideas out there to work from. Many polysomnograms include a sound recording for reference so there is a lot of data available and in the most part it is anonymised so there is not a risk to patient confidentiality. In some cases it has been labelled so it can be used easily for a retrospective study such as this one.

5.2.4 Questionnaire

A questionnaire is pretty much the only way to assess daytime sleepiness, it can also be used to enquire about other symptoms and co-morbidities. If a pre-existing questionnaire is used then there will be data on its effectiveness and usability and statistics it has produced. Questions form part of the diagnostic pathway so it takes some of the work away from the doctor. Some studies show patients are more truthful with anonymous questionnaires than with their doctor.

5.2.5 Conclusion

Given that audio analysis and questionnaires should be sufficient to diagnose OSA and rule out other similar disorders, these will be investigated further with the other methods discarded unless significant issues are encountered with audio and questionnaire analysis.

5.3 Questionnaire selection

The key point that needs to be established via the questionnaire is whether the patient suffers from daytime sleepiness however knowing about the patients weight can be useful to eliminate central sleep apnoea, and other questions about symptoms can be useful to reinforce diagnosis.

5.3.1 Own vs. pre-existing

There is a key decision between two options at this stage, to create a new questionnaire from scratch or to use a pre-existing questionnaire. A new questionnaire gives much more freedom as to what questions to ask in order to target particular symptoms and eliminate other disorders. It also allows for unique methods of analysis e.g. weighted questions rather than a simple threshold based on number of questions answered. Rigorous market research can be used to design the wording of the questions to maximise truthfulness and ease of interpretation, as well as number of questions and mechanism of answering. However this would be time consuming and an expert is question production is potentially needed. Data from patients would need to be collected via the questionnaire and other means in order to test its effectiveness, i.e. it could not be used as a retrospective study.

Using pre-existing questionnaire allows for quicker set up time as there will either be data available in order to validate the questionnaire or it will have already been validated. Some questionnaires are within the doctors guideline so time can be saved during consultation if they are already performed. Named questionnaires are often recognised by doctors which saves them time checking the questions and they are also more likely to trust questionnaire they know.

Given the time constraints and the access to data and skills further investigation will look at pre-existing questionnaires.

5.3.2 How truthful are people

One of the benefits of performing questionnaires on the app rather than waiting for patients to attend an appointment is saving doctors time however there is another benefit in the form of increasing truthfulness.

Castelo-Branco et al found that patients tended to lie in order to look good, in the context of sexuality, those involved had a strong interest in not giving a disappointing impression [10]. Rogers looked at doctors trust in patients which is an indicator of suspected lying [78]. Ulterior motives were thought to have a significant influence on patients truthfulness. In the case of OSA there is a risk of patients playing down

daytime sleepiness in order to avoid diagnosis if they wish to avoid having to contact the DVLA and their car insurance company and risk losing their license or insurance due to misunderstanding. There is also a case that patients may try and get a diagnosis in order to receive disability related financial support, however tricking a night time study into thinking you have OSA would be challenging.

5.3.3 Questionnaires

There are a number of questionnaires that have been used to assess diagnosis of OSA, some rely on a doctor to measure physical features, others use characteristic clinical features, others use patients interpretations of their symptoms, and the rest use combinations of the above.

- Chung et al STOP Questionnaire

  Four questions on snoring, daytime sleepiness, witnessed apnoeas and blood pressure [15].
- Chung et al STOPbang Questionnaire

  This is an extension of the STOP questionnaire that adds four additional questions on BMI, age, neck circumference and gender. A threshold of 3 out of 8 is generally used to indicate OSA [14].
- Chung et al ASA

  Three categories are used to ask questions, physical characteristics, observed sleep disturbances and tiredness. It uses falling into two or more of these categories as an indicator of OSA [16].
- Dixon et al

  Looked at witnessed apnoeas, neck circumference and BMI. Study rather than a questionnaire [18].
- Flemons et al Flemons screening tool

  This is a 36 question screening tool for OSA that uses a differential method for diagnosis, asking questions about depression and chronic diseases as well as the more common questions on tiredness, snoring and driving behaviour. Uses a weighted score called SACS with a threshold to indicate OSA [29].
- Kirby et al Artificial Neural Nets

  Uses 23 clinical variables including patients history, physical examination and patient reported sleepiness and smoking [43].
- Kushida et al Kushida Index

  Complicated morphometric model, including BMI, neck circumference, palatal height and oral cavity measurements amongst others. Suffers from being too complicated to be administered accurately and being time consuming [45].

- Maislin et al

  Model incorporates snoring, gasping at night, witnessed apnoeas, age, gender and BMI. Multiple logistic regressions were used to generate a multivariable apnoea risk register when compared to RDIs from polysomnographs. ROCs were used to test predictive ability [17].

- Netzer et al Berlin Questionnaire

  Uses ten questions about snoring, witnessed apnoeas, daytime sleepiness, sleeping while driving, high blood pressure and BMI. Analysis is via simple scoring [55].

- Rosenthal and Dolan Epworth Sleepiness Scale

  A questionnaire using eight questions to assess sleepiness in different situations. Recommended by NHS guidelines [79].

- Tsai et al - Upper Airway Physical Examination Protocol

  Looked at six parameters, three clinical symptoms; snoring, witnessed apnoeas and hypertension and three measurable signs; cricomental space, pharyngeal grade and overbite (info on these to follow) [89].

- Viner et al

  Model incorporates snoring, BMI, age and gender. Used stepwise linear logistic regression [92].

5.3.4 Narrowing Down the Studies

There are too many questionnaires to look at in detail however given they are not all suitable for use in a mobile app they can be reduced in number and those that are more appropriate assessed. The Kushida Index and ASA questionnaire are complicated which goes against the design specification, so can be discounted. Table 5.4 shows the statistics the studies found for each questionnaire. A high negative predictive value is needed so the focus of more work will be on; Artificial Neural Net, UAPP and the STOPbang questionnaire. Epworth Sleepiness Scale will also be investigated for use in the app because it is part of the diagnostic pathway as laid out by the NHS.

5.3.5 Artificial Neural Net

The 23 clinical variables which form parameter for the neural network are shown in Table 5.5, the patient would be aware of the demographics and social history information however night time symptoms and bed partner observations would be more challenging and impossible for the patient to know about themselves in some cases. Past medical history, anthropometrics and the physical examination in most cases the

Study	Test	n	Sensitivity	Specificity	Likelihood Ratio	Negative predictive v
Chung et al	STOP	177	74	53	1.57	76
Chung et al	STOPbang	177	93	43	1.63	90
Chung et al	ASA	177	79	37	1.25	73
Dixon et al		99	96	71	3.31	
Flemons' et al	Flemons' screening tool	180			5.17	
Kirby et al	Artificial neural net	405	99	80	4.95	98
Kushida et al	Kushida index	300	98	100	large	88.5
Maislin et al		427	60			
Netzer et al	Berlin questionnaire	744	86	77	3.74	
Rosenthal	Epworth sleepiness scale	268	76	31		
Tsai et al	UAPP	75	40	96	10	100
Viner et al		410	94	28	1.31	

Table 5.4: Questionnaire Statistics based on an AHI ¿15 threshold.

patient would not be able to self administer the questionnaire, negating its use.

Variables	Characteristics
Demographics	Age, gender
Night time symptoms	Frequent awakening, experienced choking
Bed partner observations	Witnessed apnoeas, observed choking
Daytime symptoms	Reported excessive daytime sleepiness, Epworth sleepiness scale
Past medical history	Hypertension
Social history	Alcohol consumption, smoking in pack-yr
Anthropometrics	Height, weight, BMI, systolic BP 140, diastolic BP 90
Physical examination	Tonsillar enlargement, soft palate enlargement, crowding of the oral pharynx
Clinical score	Sum of the clinical scores for the binary categorical values (maximum score = 12)

Table 5.5: Patient Demographics and Clinical Features Used in the Neural Network of Prediction of OSA

5.3.6 Upper Airway Physical Examination Protocol

This is only based on three features as shown in the flowchart(Figure **??**: cricomental space, pharyngeal grade and overbite. All of these are physiological features hence their success at diagnosing OSA. However because they are physiological features there is a chance the patient would make significant errors while undertaking the readings. Cricomental space is the perpendicular distance from the cricomental line (which joins the cricoids cartilage in the neck to the chin) to the skin of the neck (see Figure **??**), this is really hard to measure on yourself. Pharyngeal Grade measures how much the tonsils and other tissues at the back of the throat obscure the airway(see Figure **??**), this is generally graded by eye and therefore might be possible for a patient to self assess. Overbite is where the top teeth are significantly in front of the bottom teeth; this is something a patient is generally aware of. When Tsai et al studied the UAPP as

a diagnostic tool they had two different specialists assess the patients which reduces confidence in patients own ability to undertake such assessments.

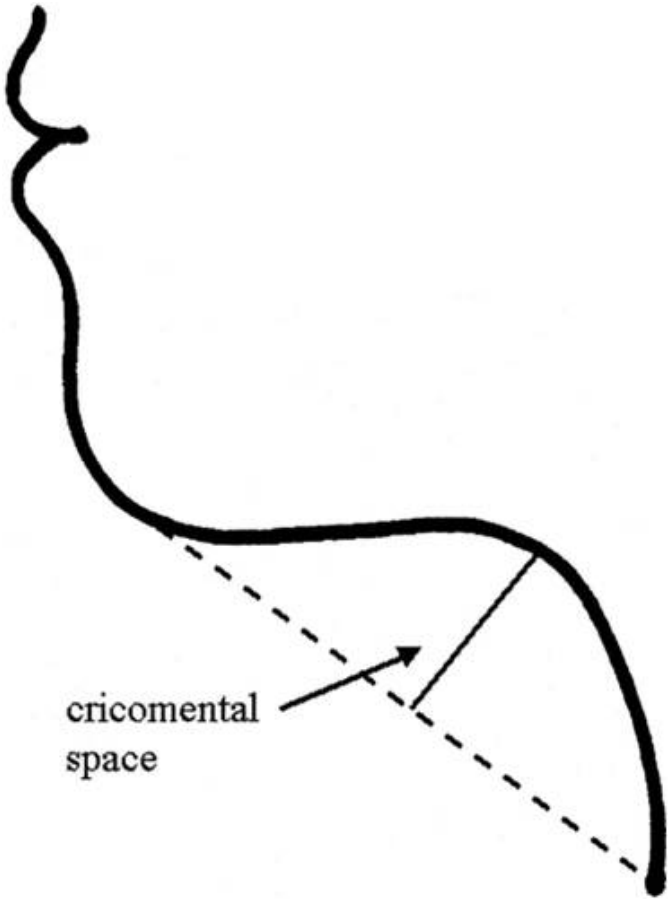


Figure 5.1: Assessment of the cricomental space. Use a thin ruler to connect the cricoid cartilage to the inner mentum. The cricomental line is bisected, and the perpendicular distance to the skin of the neck is measured [89].

5.3.7 STOPbang Questionnaire

This questionnaire is based on eight questions to do with patients self reported symptoms, each is answered with yes or no and the score is just a summation of the yes answers. Chung et al found that it was particularly good at distinguishing between severity of OSA however as a simple diagnostic tool to find everyone who suffers it is relatively effective when used with a cut off of scoring 3 or above, with a probability of 93% for detecting moderate OSA sufferers with AHI$\geq$15. 1. Snoring: Do you snore loudly (loud enough to be heard through closed doors)? 2. Tired: Do you often feel tired, fatigued, or sleepy during daytime? 3. Observed: Has anyone observed you stop breathing during your sleep? 4. Blood pressure: Do you have or are you being treated for high blood pressure? 5. BMI: BMI more than 35kgm^2?

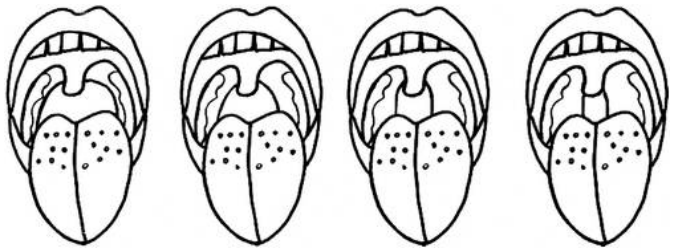


Figure 5.2: Pharyngeal grading system. Class I = palatopharyngeal arch intersects at the edge of the tongue. Class II = palatopharyngeal arch intersects at 25% or more of the tongue diameter. Class III = palatopharyngeal arch intersects at 50% or more of the tongue diameter. Class IV = palatopharyngeal arch intersects at 75% or more of the tongue diameter [89].

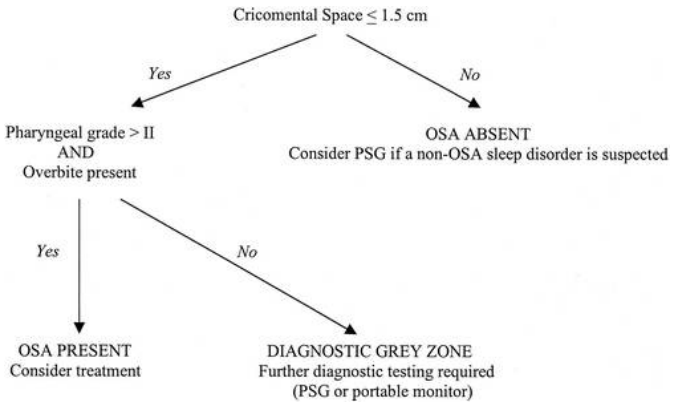


Figure 5.3: A decision rule for diagnostic testing in OSA [89].

6. Age: Age over 50 yr old? 7. Neck circumference: Neck circumference ¿40 cm? 8. Gender: Male?

5.3.8 Epworth Sleepiness Scale

This is a tool for establishing how much the patient feels they suffer from daytime sleepiness. The patient is asked to say how likely they are to doze in eight situations with 0 being they would never doze, through slight and moderate to high chance of dozing given a 3. A cut off value of 8 has been proposed by Rosenthal and Dolan which gives a sensitivity of 76% for an AHI $\geq$5. Sitting and reading Watching TV Sitting, inactive in a public place (e.g. a theater or a meeting) As a passenger in a car for an hour without a break Lying down to rest in the afternoon when circumstances permit Sitting and talking to someone Sitting quietly after lunch without alcohol In a car, while stopped for a few minutes in traffic

5.3.9 Conclusion

The STOPbang questionnaire and Epworth Sleepiness Scale are the only tests that a patient could reliably self administer, due to their shortness of length, ease to understand by a patient and ease to interpret by an algorithm, both will be included in the app.

5.4 Audio

5.4.1 Scoring of apnoea R&K vs. AASM

Audio signals taken during polysomnograms are currently scored by sleep specialists, there are two guidelines as to how to do this. The Rechtschaffen and Kales Manual is long standing, it was the only manual in use between 1968 when it was created and 2007 when the AASM Manual came into use.

The R&K manual is based on healthy subjects aged 21 to 86 years and does not actually mention how to recognise apnoeas [75]. It has also been criticised for being open to interpretation. The AASM manual has a slight change in terminology and changes distribution of NREM sleep stages but the main difference is it explains how to classify more sleep abnormalities including apnoeas and arousals [52].

The AASM scores a signal as an apnoea if: there is a drop in the peak thermal sensor excursion by more than 90% of baseline, the event last at least 10 seconds and at least 90% of the events duration meets the amplitude reduction criteria for apnoea. Obstructive apnoeas are associated with continued or increased inspiratory effort through the period of absent airflow. A minimum desaturation criterion is not required. The basis for scoring arousals is based on EEG and EMG and therefore is not helpful in terms on how to analyse audio signals [39].

Both of these guidelines are used by modern papers and so awareness of both is needed, given both result in scoring of apnoeas, choking events and snores they are useful as they lead to labelled signals which can be used for training and testing data sets.

5.4.2 Methods of analysis

The aim of the audio analysis is to find and count the apnoea-choking events to directly determine AHI, or use the difference in snoring type between OSA sufferers and simple snorers as an indicator of AHI. The initial process of this in the majority of the analyses is to split the signal down into intervals of time of somewhere between 10 and 30 seconds and assess the features of those intervals independently to determine whether they contain apnoea, choking, snoring, breathing or something else.

The starting point for assessing the intervals will be speech analysis as snoring and speech are both generated in the vocal tract although snoring is caused by vibration of the pharyngeal structures during inspiration whereas speech is vibration of the vocal cords during expiration. The digitalisation and filtering of the signals will be mainly performed by inbuilt phone software, so the focus here will be on which feature to extract, how to extract them and classification of the features.

5.4.3 Frequency of Prominent Peaks

Using Fast Fourier Transforms a power spectrum can be created of the snores. This can then be characterised in a number of ways including establishing: Fa the fundamental frequency; Fo the lowest frequency, Fpeak the peak with the maximum power, Fmean the statistical mean frequency, and Fmax can also be calculated, however there are different conventions for this, it is defined as the frequency beyond which the signal amplitude has dissipated to less than a percentage of its peak power, some studies use 3%, others 10%.

Perez-Padilla et al used Fa, Fo, Fpeak and Fmax to examine the differences between nine OSA sufferers and ten simple snorers. Defining Fmax as dissipating to 3% of peak power. Sound was recorded via a microphone attached to the manubrium sterni (chest). Significant variation was found between snores in a given patient making it hard to find a differentiator between OSA sufferers and simple snorers. It was found that for OSA suffers Fpeak was usually at a higher frequency than Fa, however this was not significant enough to be used to differentiate the two groups [94].

Fiz et al used Fpeak, Fmean and Fmax defined at 10% of peak power to distinguish between ten OSA sufferers and seven simple snorers. The microphone was placed just above the larynx without skin contact. Two features were found to distinguish OSA sufferers from simple snorers. The first was peak frequency (Fpeak) which was significantly lower in OSA sufferers with a threshold of 150Hz, although one of each group was on the opposite side, a strong non linear negative correlation was seen between Fpeak value and number of AHI events as seen in Figure **??**, this is associated with a Spearman rank order correlation: r=-0.70; p¡0.0016. The second feature was to do with the shape of the power spectrum. The simple snorers displayed a clear fundamental frequency and harmonic pattern whereas the OSA sufferers displayed a low frequency peak with scattered peaks over a narrow band of frequencies, Figure **??** shows the different patterns. Fiz et al attributed the differences seen to microphone placement, Perez-Padilla et al placing their microphone on the chest whereas Fiz et al placing above the larynx, this results in a different filtering effect caused by the different tissues and cavities the sound travels through [28].

Prominent peaks alone are not sufficient to diagnose obstructive sleep apnoea.

5.4.4 Power Ratio

Given the difference in spectra between those with OSA and simple snorers there is potential to characterise this by means of a cumulative power ratio. This would be the area under one part of the power

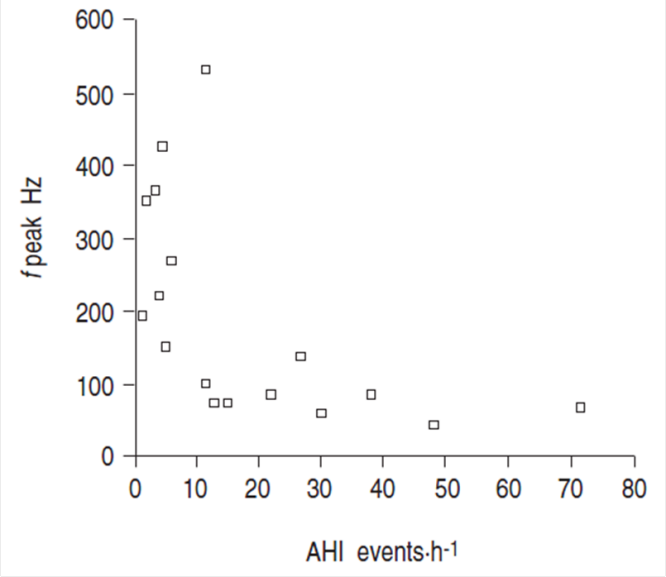


Figure 5.4: Relationship between apnoea/hypopnoea index (AHI) and peak frequency (Fpeak) of spectrum in seven simple snorers and 10 obstructive sleep apnoea (OSA) patients. There is a significant negative correlation (Spearman rank order correlation: r=-0.70; p¡0.0016) [28].

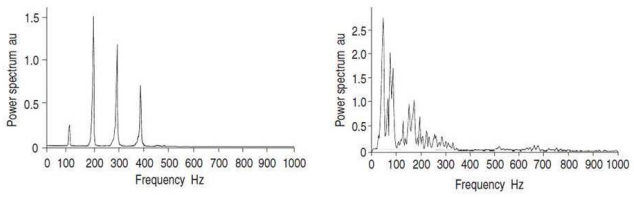


Figure 5.5: Power Spectrum of first breath from a simple snorer and OSA patient [28].

spectrum divided by the area under the rest, where the threshold is placed would depend on what characteristic was trying to be differentiated.

Perez-Padilla et al used superimposed spectra from ten snores from each subject (9 OSA, 10 simple snorers) and a threshold of 800Hz, dividing the integral of the spectra above 800Hz with that below 800Hz, for OSA patients who took a second breath after an apnoea the cumulative power ratio was also calculated for that. Figure **??** shows the scattering of the ratios, with simple snorers having a ratio of 0.08±0.02 and OSA sufferers a ratio of 1.12±0.31. A threshold of 0.3 was proposed which would distinguish all but one OSA sufferer on first breath after apnoea, this patient had a low AHI and mild symptoms [94]

Hara et al used the same method as above except power ratio was defined as the inverse on 46 OSA sufferers and 12 simple snorers, keeping the threshold at 800Hz, using a ratio of below 800Hz to above 800Hz. Simple snorers had a power ratio of 34.002 (0.03 when defined as in the Perez-Padilla study) whereas OSA sufferers had a ratio of 6.288 (0.16 when defined as in the Perez-Padilla study). The p value of a Mann-Whitney U test was 0.015, this small value is a good indicator that this did not happen by

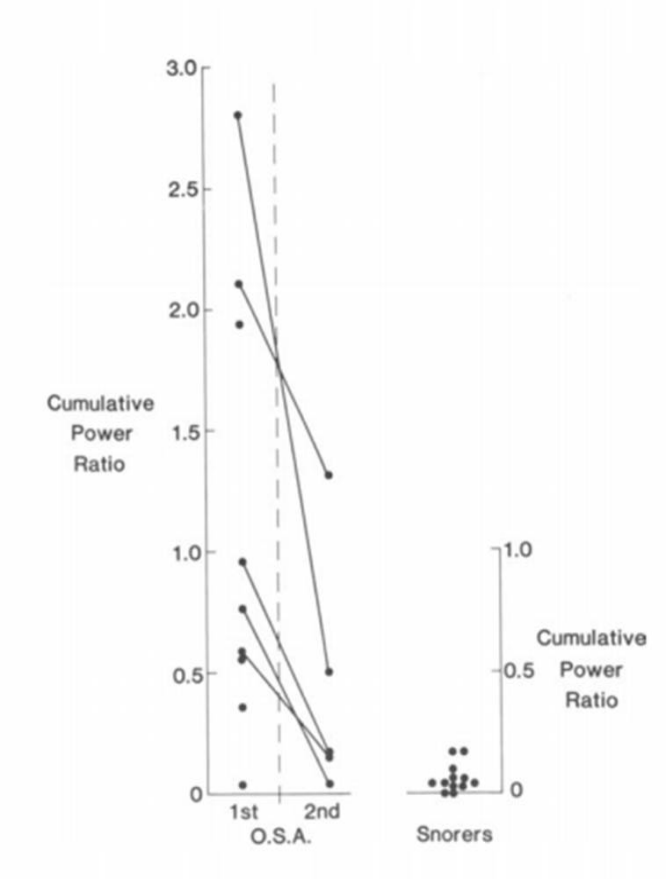


Figure 5.6: Cumulative power ratio between frequency components higher and lower than 800 Hz in patients with OSA and in snorers. [94]

chance [37].

Hara et al would not recommend this approach as the calculation is very time consuming [37]. Significant differences were found between the values of the ratios in both studies although in both cases a difference between OSA sufferers and simple snorers was clearly seen.

5.4.5 Sound Intensity

Sound intensity can be used for diagnosis in a couple of ways; as a cut off got distinguishing between snores and apnoeas in order to look for apnoea-arousal events, and as a distinguisher between apnoeic snores and non-apnoeic snores. In both cases frequency need not be known although in the first approach the signal does need to be split into temporal intervals.

Van Brunt et al defined an acoustic signature event as an apnoeic event lasting between 10 and 90 seconds followed by snoring. Where apnoea had an intensity below 50μV and snoring above 100μV. 30 second intervals were analysed by sound intensity and polysomnogram and considered in two ways: firstly

each patient (69 patients, 51 OSA suffers, 18 not) was assessed independently and RDI score compared with predictions, for an RDI of 15/hour or greater sensitivity was 93%, specificity 25%, false positive 36.2%, false negative 2.8%, 60.9% classified correctly. Secondly pooling all the observations (60231) resulting in a 33% sensitivity, 98% specificity, and 85% classified correctly [90].

Wilson et al looked at four sound intensity levels: L_1, L_2, L_3 the dB levels exceeded for 1%, 5% and 20% of the test period respectively, and L_{eq} the average dB level. 1139 patients were studied (682 OSA, 261 simple snorer and 196 unknown), differences were seen between the sound intensity of apnoeic and non-apnoeic snores with non-apnoeic snores having a lower intensity as seen in Table 5.6. A logistic regression model with RDI ≥ 10 as the dependent variable found a cut off of $L_{eq} \geq 38$ to be significant, with a regression coefficient of 1.24, a standard error of 0.28 and an odds ratio of 3.44 (95% CI) [100].

Sound Intensity Measure	Nonapnoeic Snoring (RDI ¡ 10)	Apnoeic Snoring (RDI $\geq$ 10)
L_{eq}, dBA	42.7 (42.043.4)	48.8 (48.749.3)
L_1, dBA	53.4 (52.854.1)	59.2 (58.759.7)
L_5, dBA	48.0 (47.448.6)	53.2 (52.753.7)
L_{10}, dBA	46.0 (45.546.5)	50.4 (49.950.9)

Table 5.6: Sound Intensity Measures for Subjects With Apnoeic Snoring (RDI $\geq$ 10) and Non-apnoeic Snoring (RDI ¡ 10).

5.4.6 Formants

Formants are the resonant frequencies of the signal, most easily determined by picking out the peaks on a linear predictive coding (LPC) spectrum of the signal. LPC is the spectral envelope produced by a linear predictive model. The lowest formants are associated with degree of constriction of the pharynx, degree of advancement of the tongue and degree of lip rounding respectively, these are often referred to as F1, F2 and F3. Because these physical properties change in sufferers of OSA there is a chance that the frequencies associated with them will change too, therefore threshold frequencies to distinguish OSA sufferers and simple snorers are sought.

Sola-Soler et al used formant frequencies to distinguish between snores from eight simple snorers (447 snores) and eight OSA sufferers (236 normal snores and 429 post-apnoeic snores). The spectral envelope was estimated by linear predictive autoregression, with very low amplitude spurious peaks rejected by a 3dB threshold. Investigation of the spectral envelope found 2 to 6 formants in each snore, these fell in common frequency range around 150Hz, 500Hz, 1KHz, 1.7KHz, and in a few snores 2.2KHz. This led to definition of five frequency bands: B1[0,300), B2[300,700), B3[700,1400), B4[1400,1900) and B5[

1900,2500) in Hz. For each band and type of snore (simple snorer SN, OSA normal snore OP-N, and OSA post apnoeic snore OP-PA) the mean value Fi and standard deviation SFi was calculated. Table 5.7 shows a comparison between these means and standard deviations for each combination of snore type calculated using the Mann-Whitney U test, the fifth band was left off because so few snores exhibited this formant. If a snore had more than one formant in a band the average frequency of those was used in the calculation. Bands 1 &3 showed significant differences between formants when comparing the standard deviation of simple snorers and OSA sufferers both in normal snores and post apnoeic snores of the OSA sufferers. The most distinct difference was between the standard deviation of simple snorers normal snores and post-apnoeic snores in band 1 which had a probability of 0.0006 [84].

Contrasted populations	F1	SF1	F2	SF2	F3	SF3	F4	SF4
SN 1 OP-N	0.4179	0.0151	0.0882	0.0618	0.1556	0.0045	0.5839	0.8551
SN / OP-PA	0.7104	0.0006	0.1061	0.6389	0.5338	0.0012	1	0.1255
OP-N 1 OP-PA	0.6434	0.0372	0.4822	0.0842	0.5628	0.203	0.631	0.1495

Table 5.7: Bilateral significance in Mann Whitney U test to contrast population difference.

Ng et al tested 30 OSA sufferers and ten simple snorers and found a threshold for F1 of 470Hz but not F2 nor F3, this threshold yielded a sensitivity of 88% and a specificity of 82%. However increased sensitivity and specificity were seen if different thresholds were used for men and women as seen in Table 5.8. Women show a greater distinction in F1 frequency than men as well as a reduced spread of results, although more outliers amongst the simple snorers, as seen in Figure **??**. It is however worth noting that the sample size for women was small (6 OSA sufferers, 4 simple snorers) Once the threshold has been established an equation is needed to convert it to an AHI value, a number of equations were proposed and regression used to see which was the best fit, as seen in Table 5.9. A power law came out best with a regression of 0.5334 giving a predicted AHI score of 12.2 when 10 was being aimed for. It is worth noting that the same subjects was used for training and testing, although different data for each [57].

Type	AHI	Data size (train ; test)	F1	F2	F3	Threshold of F1 (AUC : F1; F2; F
M:A	52.8±24.7	720 ; 240	750±204	1842±238	3008±266	545
M:B	5.3±3.5	180 ; 60	425±157	1670±271	2858±405	(0.8977; 0.6860; 0.6117)
F:A	23.6±15.1	180 ; 60	623±150	1676±303	2739±282	298
F:B	3.4±3.2	120 ; 40	263±187	1564±294	2813±297	(0.8969; 0.5894; 0.5850)
C:A	46.9±25.7	900 ; 300	724±201	1809±261	2955±290	470
C:B	4.6±3.4	300 ; 100	360±187	1627±285	2840±366	(0.8992; 0.6820; 0.5964)

Table 5.8: AHI, apnoeahypopnoea index (events/h); Data size, number of snores; AUC, area under receiver operating characteristic curve; Sens, sensitivity (%); Spec, specificity (%).

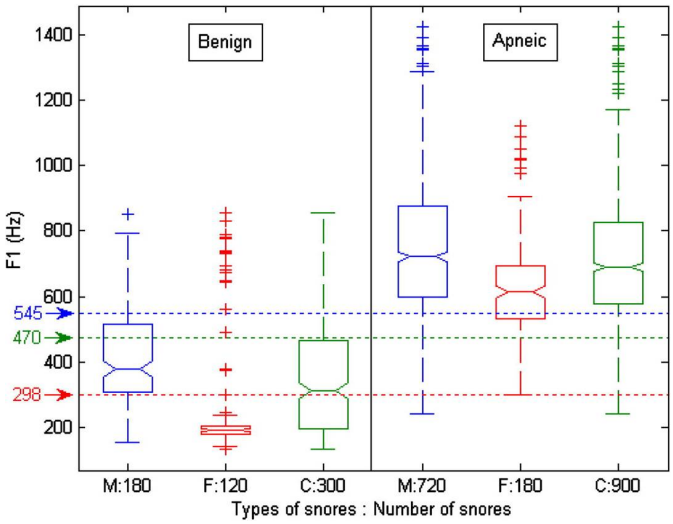


Figure 5.7: Notched box plots of apnoeic and benign snores in the training dataset for males (M), females (F), and for both males and females combined (C), under first formant frequency (F1) analysis, with marked threshold values, from Table 5.8. [57]

Regression	Equations	R2	Predicted AHI/h
Linear	AHI = 0.0738 (F1) 10.4067	0.3449	24.3
Quadratic	AHI = 0.0001 (FI)2 + 0.1778 (F1) 39.2095	0.3824	26.3
Logarithmic	AHI = 212.4891 + 89.9464 Log (F1)	0.3553	27.9
Exponential	Log (AHI) = 0.1545 + 0.0018 (F1)	0.4602	10.3
Power	Log (AHI) = 5.2100 + 2.3565 Log (F1)	0.5334	12.2

Table 5.9: Regression analysis for apnoeahypopnoea index (AHI) in events/h and first formant frequency (F1) in hertz
AHI computed at F1 = 470 Hz. Ideally predicted AHI = 10 events/h.

Yadollahi et al used formant frequencies to distinguish between snores and breaths rather than simple snorers and OSA suffers so a direct comparison of method cannot be made however there is value in exploring the method used. Bands were used as in the Sola-Soler study but at different frequency ranges, [20400]Hz, [270840]Hz, [5001380]Hz, [9101920]Hz, [1680-2680]Hz, [2580-3770]Hz and [3590-5000]Hz these were found using Kmeans clustering. This is a method of partitioning data unsupervised, it uses an iterative method to find a predefined number of partitions, in this case seven, an error margin of 10^5 was used for the iterations. The process is sensitive to initial conditions so was repeated 20 times with different initial conditions, the one with the minimum error was selected. Table 5.10 shows the student t-test p values, which show the first and third formants to be significant, with probability p = 0.003 and p = 0.0244 respectively. The F1 frequency of the breath sound was greater than that of the snore while the F3 frequencies were the converse [102].

Formants	F1	F2	F3	F4
p-value	0.0003	0.1793	0.0244	0.7009

Table 5.10: Results of P-values of t-test between formants of snore and breath sounds. represents the significant values.

5.4.7 Sub-band Energy Analysis

This uses the vertical box control chart to segment the signal temporally into intervals, this is a non-parametric detection rule which uses a moving vertically trimmer box, hence the name [72]. The ten dimensional 500Hz sub-band energy features are computed and then reduced to two dimensional feature space by principal component analysis, this is a statistical procedure that uses orthogonal transformations, this results in the largest possible variance within the first component. Classification can then be performed on this 2D feature space to separate the desired classes.

Cavusoglu et al computed the normalised 500Hz sub-band energy distribution for ten apnoeic snorers and eighteen simple snorers and used principle component analysis to reduce it to two dimensional feature space in order to find the most discriminant features. Robust linear regression was used to separate snores from non-snores. This had a sensitivity of 89.3% and a specificity of 96.3% [11].

Azarbarzin et al used a similar method to Cavusoglu except that K-means clustering was used as the classifier for the thirteen apnoeic snorers and seven simple snorers. When the classification was compared to manually annotated signals, a sensitivity of 94.8% and a specificity of 96.3% was found for the distinction between snores and non-snores in OSA sufferers [5].

5.4.8 Bispectral Analysis

Bispectral analysis exploits the fact that the bispectrum reveals both amplitude and phase information about a spectrum, while also being calculated by convolution makes it the easiest polyspectra to compute. Polyspectra are Fourier Transforms of cumulants, for example the second order cumulant; autocorrelation Fourier Transforms to the Power Spectrum. The third-order cumulant (**??**) transforms via a Double Discrete Fourier Transform (DDFT) to the bispectrum (5.2). Although the bispectrum is often plotted on a square it is symmetric and so only a triangular region is needed to completely describe it, this is defined by$0 \leq \omega_2 \leq \omega_1, \omega_1 + \omega_2 \leq \pi$.

$$C(m,n) = E\{x(k)x(k+m)x(k+n)\} \tag{5.1}$$

$$B(\omega_1, \omega_2) = E\{X(\omega_1)X(\omega_2)X^*(\omega_1 + \omega_2)\} \tag{5.2}$$

Quadratic phase coupling (QPC) occurs uniquely in second-order non-linear systems, and is where the phases add and subtract along with the frequency components [74]. QPC causes peaks in the bispectrum triangular region, this shows energy is produced at frequency $\omega_1 + \omega_2$, a flat bispectrum at ω_1 and ω_2 suggests no activity and that it is not affected by QPC.

Ng et al exploited the bispectrum shape in order to distinguish between nine OSA sufferers and seven simple snorers. The bispectrum was plotted (Figure **??**) and inspected visually, the axes have been normalized with a frequency of 1 being 11025Hz. From visual inspection is appears the biggest peak for simple snorers are near the origin while for apnoeic snores are further away, this suggests a greater degree of phase coupling in apnoeic snores due to nonlinearities in the signals. When analysed peak position was reflected in the numbers with apnoeic peaks are higher frequencies than simple snore peaks, Table 5.11 shows the figures [60].

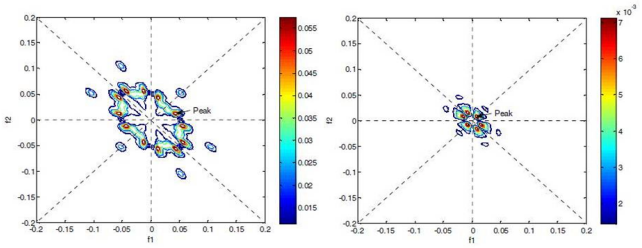


Figure 5.8: Bispectrum of a common benign snore [60]

Types of snores	f1(Hz)	f2(Hz)
Apnoeic	233 ± 147	170 ± 82
Benign	154 ± 51	140 ± 61

Table 5.11: Mean peak frequencies of apnoeic and benign snores

Clustered multiple comparison graphs were then produced, Figure **??**, QPCs appear mainly at f1=f2 for simple snores whereas they appear mainly at f1≠f2 for apnoeic snores, however there is less self coupling. 77% of simple snore are self coupled compared to 49% of apnoeic snores. So the analysis shows there are three differences between simple snores and apnoeic snores, apnoeic snores have strong presence of nonlinear interaction, less self-coupling and QPC peaks appear at higher frequencies.

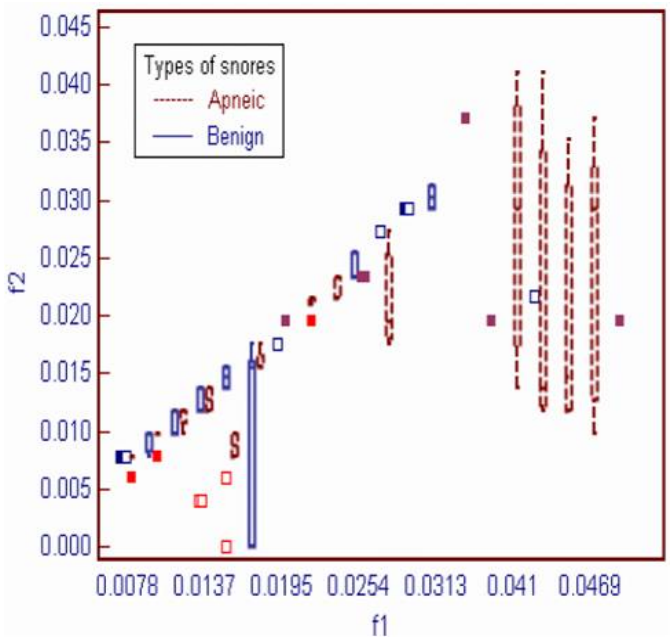


Figure 5.9: Clustered multiple comparison graphs. [60]

5.4.9 Wavelet Analysis

Wavelet analysis is based on basis functions much like Fourier transforms (they do not have to be orthogonal although they often are), however wavelets are used rather than sines and cosines. As wavelets decay rapidly with time they can be used as a function of time as well as frequency.

Ng et al in 2008 used a special type of discrete wavelet transform that was translation which is a non-orthogonal and undecimated adaption of the former, solving a number of fundamental issues it suffered mainly to do with sensitivity with alignment in time, and Gibbs like artefacts around discontinuities, to distinguish between 30 OSA sufferers and ten simple snorers. Ten snores of about 6 seconds were randomly chosen from each snorer, marked by polysomnogram technologists. Figure 5.10 compares the energy approach used by Cavusoglu et al and the time invariant discrete wavelet transform (TIDWT) approach used by Ng et al in this study. At the narrowest tolerance of 25ms the TIDWT approach detected 51% of the snore segments correctly compared with 8% for the energy approach, and at the largest tolerance of 125ms the TIDWT approach detected 98% correctly, whereas the energy approach only managed 87%, i.e. the TIDWT approach is at least 10% more effective [59].

Ng et al in 2009 used continuous wavelet transform in the form or wavelet bicoherence analysis. Bicoherence is the normalised bispectrum and a measure of the amount of phase coupling in a signal [91]. Apnoeic snores showed phase coupling at higher frequency modes than benign snores, with apnoeic snores being 20% less self coupled that benign snores. Two markers were proposed, these were peak frequency

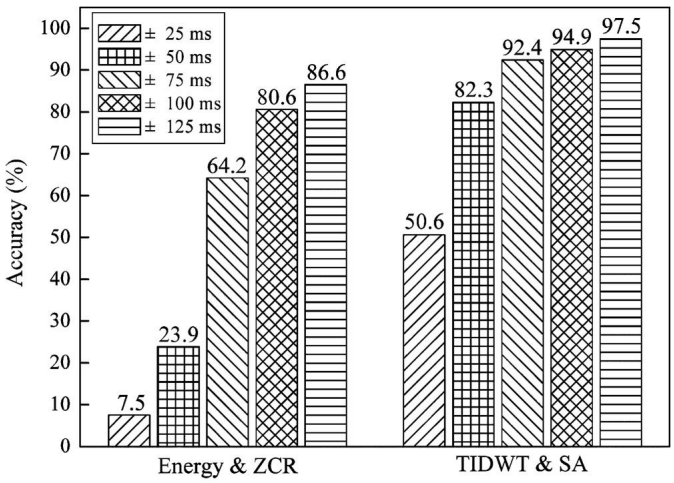


Figure 5.10: Accuracy comparisons between conventional- (energy and ZCR) and wavelet-based (TIDWT and SA detector) approaches in snore segment boundary detection at several tolerance degrees: 25, 50, 75, 100, and 125 ms. [59]

component at f_1 (PF1) and peak sum frequency (PSF) i.e. the frequencies with the highest coupling strength of frequency 1 and the sum of frequencies 1 and 2. The relationship between the markers and AHI was assessed using various regression models the most successful of which were exponential and power models using residual standard deviation statistic. These produced optimal thresholds for the markers of PF1 = 285 Hz and PSF = 492 Hz resulting in a sensitivity of 85.0% and a specificity of 90.7% [58].

5.4.10 Multiclass Classification

Rather than looking at each potential feature separately as has been done so far it is possible to use regression analysis and Bayes classifiers (a type of machine learning), amongst others to predict AHI level from a sound signal with the need to count the number of apnoeas, instead looking at the overall shape of the signal.

Ben-Israel et al investigated the use of five classifiers on 60 training patient sets and 30 testing patient sets. The features were: Mel-Cepstability, a measure of stability of the spectrum over the whole night; running variance, a measure of the energy variability between snores over the whole night; apnoeic phase ratio, percentage time the upper airway collapsed using an energy threshold; inter event silence, a count of apnoeic events based on energy patterns; and pitch density, a measure of stability of fundamental frequencies. This resulted in a five dimensional feature vector, which was classified by a Gaussian Bayes classifier. A threshold of AHI ¿10 resulted in a sensitivity of 87% and a specificity of 80% [9].

Sola-Soler et al classified their subject into three classes C_1 simple snorers with AHI ¡5, C_2 mild to

moderate OSA sufferers 5≤ AHI ¡ 30, and C_3 sufferers of severe OSA with AHI ≥ 30. There were 13 simple snorers, 11 C_2 and 12 C_3 sufferers. 22 features were used which fell into the categories of sound intensity, pitch, frequency parameters and formants (see Table 5.12 for the full list). These were then classified in two main ways, logistic regression and Bayes rule. A forward stepwise selection algorithm was used to find optimum independent variables for two binary logistic regression classifiers used simultaneously, two classifiers were needed because there are two thresholds [85].

Three different types of naive Bayes classifier were investigated, the optimum independent variables were found by a sequential floating forward selection algorithm up to a maximum of ten variables. The three approaches were Gaussian, kernel diffusion and kernel density; with kernel density producing the best results. Gaussian produced the worst results for the training set probably because snore features are not Gaussian. Bayes using kernel density performed better than logistic regression with accuracy of 83.3% and 0% false negatives whereas logistic regression was 72.2% and 8.3% respectively [85].

[h] %width of table will need manual changing

Snore intensity parameters	
I_mean	Mean sound intensity
I_max	Maximum sound intensity
Snore pitch parameters	
P_m	Pitch mean value
P_s	Pitch standard deviation
P_{iqr}	Pitch interquartile range
P_{dens}	Pitch density
P_{ints}	Pitch intervals
Snore frequency (PSD) Parameters	
F_{mean}	Mean frequency
F_{med}	Median frequency
F_{peak}	Peak frequency
F_{max}	Maximum frequency
CSymm	Symmetry coefficient
CFlatn	Flatness coefficient
RW_B	Ratio of energy in band B to tot
$Rout_B$	Ratio of energy in band B to ene
Spectral envelope snore parameters	
F_i	Frequency of the ith formant
L_i	Attenuation of the ith formant
M_i	Amplitude of the ith formant

Table 5.12: Snore characterisation. B is the frequency band: B = (0, 500)Hz, (100, 500)Hz or (0, 800)Hz, i = 1:5.

5.4.11 Conclusion

As has been shown a number of features can be used successfully to differentiate obstructive sleep apnoea sufferers from simple snorers with sub-band energy analysis and multiclass analysis performing the best. There were three ways the signals were classified: regression, K-means clustering and Bayes classifiers. Bayes classifiers and K-means clustering consistently outperformed regression classifiers. These are types of machine learning, although not the only types to be used for the classification of signals used in the diagnosis of OSA, other types of machine learning used include; Hidden Markov Models used by Rossow et al to classify EEG signals [80]; Support Vector Machines used by Gederi and Clifford to classify images [35], and neural networks used by Roberts and Tarassenko to classify EEG [76]. Given the success of machine learning in the cases shown even with very small training and testing sets this will be the focus of the next work.

5.5 Signal Analysis – simple method

5.5.1 Outline

The simple method of diagnosing sleep apnoea, using audio data from the user's phone that is recorded while he/she is sleeping, was meant to provide a starting point for the development of the app. It served three main purposes:

- To familiarise us with the typical sleep patterns and with power and frequency content of sleep data
- As a backup

  This would give us a method that is able to provide a diagnostic output such as the apnoea-hypopnoea index score from sleep data, even if it was not using machine learning algorithms.
- As a point of comparison

  The simple method would serve as a point of comparison, along with other methods used, with regards to the accuracy of the diagnosis. Using simple metrics such as percentage accuracy of pre-determined apnoea or hypopnoea points in the data against what the model is able to pick out, we can compare the simple model results with our machine learning results. This would allow us to gauge our preformance and also prove that the app is able to diagnose OSA with superior accuracy.

MATLAB® was used as the development environment for the simple model. This was due to several reasons. Firstly, our familiarity with MATLAB® from previous projects made it a natural starting point. Secondly, as a dynamically typed language with a user-friendly interface, MATLAB® would enable us to make minor changes to the code and observe the results quicker. Experimentation and tinkering with the code is much easier in MATLAB® than it is in C/C++ or Java, for example. This ease of experimentation meant that MATLAB® is a good language to start writing the model in, and is used in the initial development of the machine learning HMM model as well. With in-built functions such as audioread, MATLAB® provided the basic tools with which we could develop the model. Of course, if necessary, the model could later be written in Java for implementation in an Android App - this would be trivial.

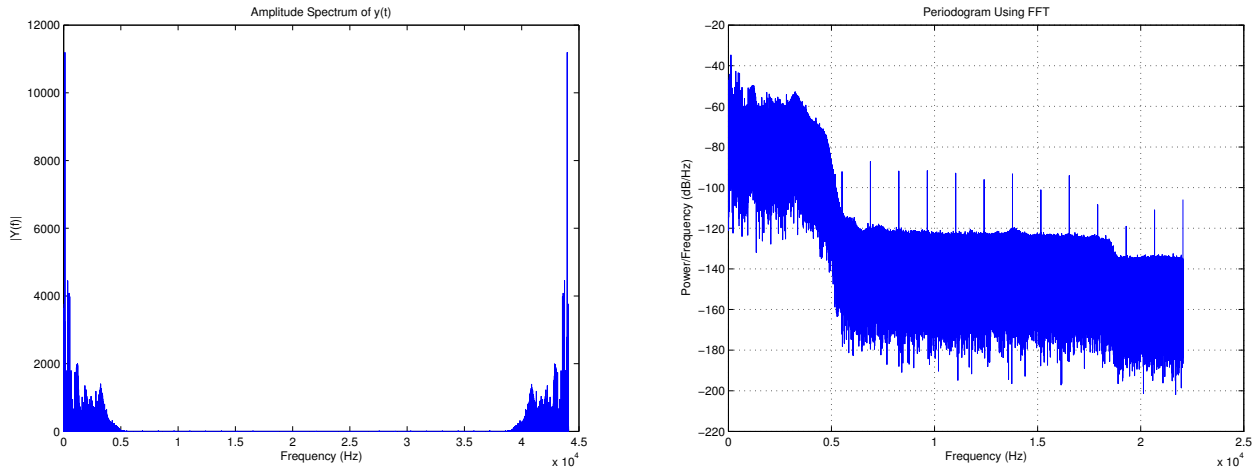
The MATLAB® code for the simple model is presented in Appendix A, with detailed explanations. A brief summary is as follows: three `.m` files are created – `simple.m`, the main script which is run once an audio file is created and stored, and two functions `detectApnea` and `detectApneaVar`, that take a

vector of the signal power and a few other parameters as inputs. The `simple.m` script uses the `wavread` or `audioread` function (the user's audio sleep data is recorded in `.wav` format) to read the file and produce a matrix of the audio level values. This is sampled at a lower frequency in order to reduce memory storage. The frequency content of the signal is calculated using the fast fourier transform function (FFT). Along with a plot of the signal, a plot of the frequency content is generated (this is done purely for the programmer's convenience and analysis). The power content is calculated from the signal vector and is used in the `detectApnea` function. An additional function, `detectApneaVar`, is included in the code below highlighting two different ways of analysing the data. The first way, used in `detectApnea`, uses a simple threshold parameter, and searches the signal power vector for prolonged periods of time when the value is below a threshold level. These periods represent apnoeatic episodes when the user's airflow experiences blockages. The second way, used in `detectApneaVar`, aims to identify the sudden inspiration that accompanies the body attempting to re-establish airflow. This is usually characterised by a snorting noise from the user. Instead of simply looking at the power values and comparing them to a threshold, `detectApneaVar` identifes periods of high variance in the signal, which represent the user being 'shocked' to start breathing again.

Now, we describe the `simple.m` script in Appendix A.1. The first part of the code assumes that the user's sleep data has already been recorded using the microphone and has been stored in a file `signal.wav`. This file is read using the `wavread` function as mentioned, and outputs `YRaw`, a two-column matrix (as `wav` encoding uses two channels) as well as `f`, the sampling frequency which is 44.1 kHz. The length of the audio file in seconds, `TMAX`, is also calculated for later use.

A sampling frequency of 44.1 kHz is unnecessarily large and wasteful for an application such as ours – the frequency content of a 7-minute sample obtained from YouTube [46] is shown below. The figure on the left represents the spectral density while that on the right represents the information on a logarithmic scale so as to highlight where the useful information is found. As can be seen, there is no useful information that can be found at frequencies above 5 kHz (the duplicated spectra on the left are a result of sampling at the frequency of 44.1kHz). There is no risk of aliasing occuring even if the audio is sampled at 5 kHz. This is where the scale parameter in line 9 above comes in – it reduces the size of `YRaw` by sampling the already sampled vector (and also combines the two channels into one by taking the average).

The frequency content shown above is calculated using the fast fourier transform (FFT) function in MATLAB®. The function calculates the discrete fourier transform (DFT) of a vector $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$



(a) Amplitude Spectra of Raw Audio File

(b) Periodogram of Raw Audio File

Figure 5.11: Frequency content of Raw Audio File

and returns output $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ such that

$$X(k) = \sum_{j=1}^{j=N} x(j)\omega_N^{(j-1)(k-1)} \tag{5.3}$$

where

$$\omega_N = e^{(-2\pi i)/N} \tag{5.4}$$

The next few lines of `simple.m` serve to calculate and plot, again, the frequency of the reduced vector `Y` (the figures above are for `YRaw`). The vector power is also created to hold values for the power of the signal at each time period, and will be used in the functions detectApnea and detectApneaVar.

The next part of the script sets some parameters and uses `detectApnea` as well as `detectApneaVar` to analyse the signal power. The results are then put together and plotted for the programmer's benefit. (The code for doing so is not shown here as it is relatively straightforward and unimportant).

The function `detectApnea` takes as inputs the vector power, the length of the audio file `TMAX`, and parameters `sensitivityMean` and `interval`.

We now describe the code for `detectApnea` shown in Appendix A.2. Taking inputs `sensitivityMean` and `interval`, `detectApnea` runs through the vector power and identifies when the signal power is below a threshold level for at least a certain interval. This threshold level can be adjusted by changing `sensitivityMean`, and also is affected by the mean value of the overall signal power. This is important,

as it is a first step towards ensuring that variations due to users putting the phone further away, which would reduce the overall power of the signal, are taken care of to some extent. The output vector apnoea contains ones and zeroes at every sample point, with ones representing apnoea or hypopnoea episodes.

We now describe the code for `detectApneaVar` shown in Appendix A.3. Instead of simply searching the vector power for values lower than a threshold, `detectApneaVar` attempts to identify periods when there is a sudden spike in the signal power, which means the user has snorted and been shocked into breathing again. Similar to `detectApnea`, `detectApneaVar` uses parameters such as `sensitivityVar` and `windowSize`.

As can be seen, a variance method is used to detect sudden spikes in the signal. A moving window is created of size `windowSize` and is used to temporarily house parts of the vector power. The variance of the values in the window is calculated and if it exceeds a value `maxVariance`, the user is recognised as having had an apnoeatic episode. The window moves on to the immediate next period after updating the output vector `apnea`, and by the end the vector `apnea` contains ones and zeros identifying points in the signal where apnoea is thought to have occurred. Once again, variations due to different users and settings have been accounted for to a certain extent by calculating the value of `maxVariance` from the variance of the entire signal.

The results from the two methods of detecting apnoea are combined together, and the results are plotted below for the sample YouTube video. Firstly, the plot of the power signal, with peaks at regular intervals, confirms that the user has OSA and experiences apnoea and hypopnoea in predictable cycles. The results from the two methods are plotted below the power signal, and exhibit enough correlation such that combining the methods can be justfied.

5.5.2 Limitations of model

While the simple model we have used above gave us a good starting point and will serve as a point of comparison, it has many limitations. Firstly, the choice of parameter values plays a huge role in the accuracy of the diagnosis. Even though some effort at reducing the effect of variations in noise and signal strength has been taken, the choice of sensitivityMean, sensitivityVar, interval and windowSize is still arbitrary to a large extent. This means that while the model works well for the sample above, there is no guarantee of its effectiveness for other audio signals.

This is where machine learning comes in - needing to choose arbitrary parameter values is removed as the program can be designed to learn, with training data, the best values for parameters and update itself

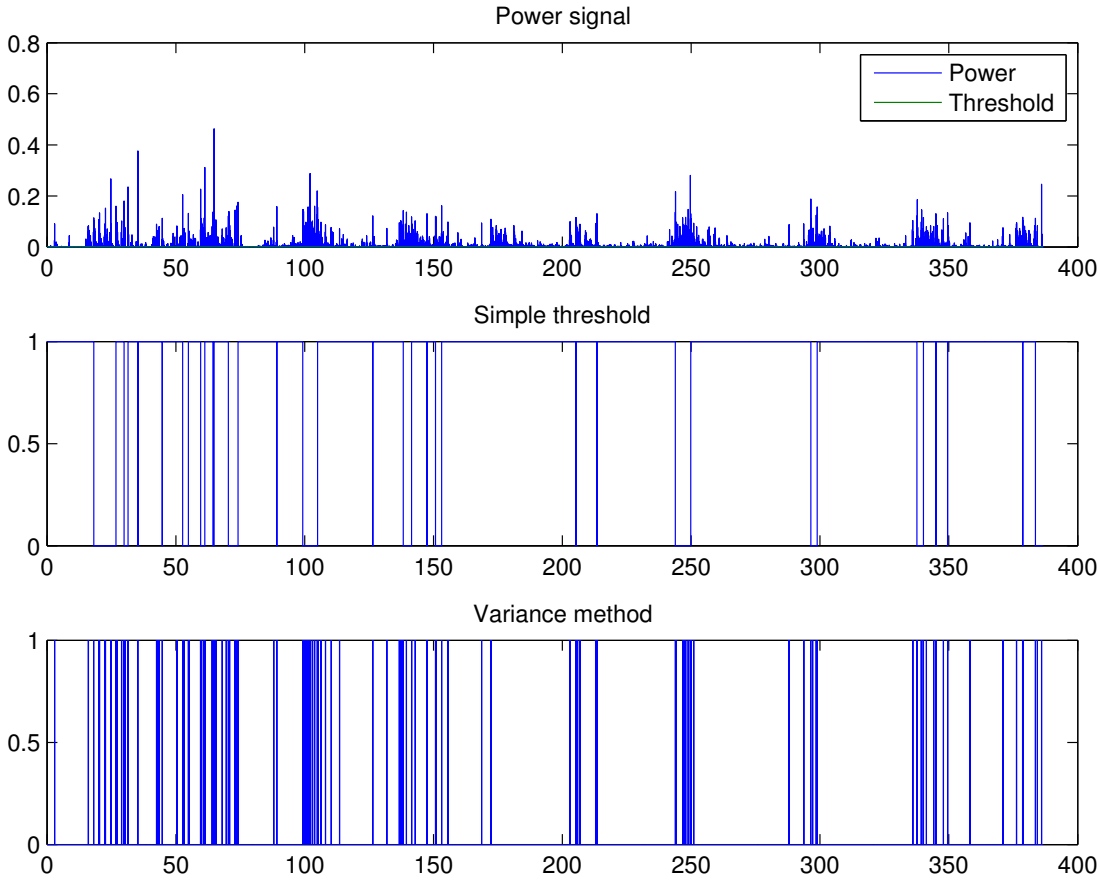


Figure 5.12: Diagnosis Results from 7-minute Sample

accordingly. The parameters do not have to be the same as those above, of course - this would depend on the exact machine learning model being used.

Secondly, the simple model assumes the existence of an audio file that has already been recorded. Given the average sleep time for adults of seven hours, this could lead to memory storage issues in smartphones where the recording is meant to take place. Some form of compression of the data needs to take place even while recording, and this issue is tackled in the following sections of the report.

5.6 Machine Learning – Theory

Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data [98]. The supervised classification problem in machine learning concerns finding the unknown target function that classifies certain input data into classes based on some set of training examples containing labelled input data.

It is exciting to use machine learning algorithms to aid medical diagnoses because it can dramatically save time for medical experts. These techniques should create a reasonably reliable, though not perfect, system on which the medical experts can base their decisions on.

In our case, given a sampled sound signal of a sleeper, we want to identify apnoea periods. In particular, we represent our input (sampled sound signal) as $\mathbf{S} = \{s_1, s_2, \ldots, s_T\} \equiv \{s_i\}_{i=1}^{T}$, and we want to output the classifiers for every K samples, $\mathbf{Y} = \{y_i\}_{i=1}^{T/K}$, where the classifier $y_i \in \{0, 1\}$ corresponds to samples $\{s_j\}_{j=(i-1)K+1}^{iK}$ of the signal. We assume that T is divisible by K, however if this is not the case, we discard an appropriate number of signal samples to fulfill this condition.

We have researched three models for our problem which we will discuss in this section. Firstly, we will discuss Support Vector Machines which are one of the most widely used algorithms in Machine Learning today, then we will discuss the State-Space Model and the Hidden Markov Models, which are more well-suited to our problem due to their temporal nature. Moreover, we discuss techniques to condition our data before using them as input data for our learning models.

5.6.1 Support Vector Machines

Here, we present the theory for Support Vector Machines (SVMs) based on the lecture notes from Prof. Andrew Ng [56]. SVMs are one of the most widely used and many argue among the best "off-the-shelf" supervised learning algorithms. This is mainly due to the sound theoretical framework, efficiency and good generalisation guarantees even for high-dimensional and linearly non-separable data.

Notation

Having m training examples, where

- $\mathbf{x}^{(i)} \in \mathbb{R}^d$ is the d-dimensional i-th training example.
- $y^{(i)} \in \{-1, 1\}$ is the i-th training label.

we want to find the parameter $\mathbf{w} \in \mathbb{R}^d$ which describes the hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$ that separates our two classes. Thus, we can define our classifier as $h_{\mathbf{w},b}(\mathbf{x}) = g\left(\mathbf{w}^T\mathbf{x} + b\right)$, such that $g(z) = 1$ if $z \geq 0$ and $g(z) = -1$ otherwise. Note that this is a non-probabilistic learning model as we are not considering the probability of each class or the data.

Objectives

The essence of SVMs lies in finding the decision boundary (hyperplane) which maximises the gap between the closest training points to it. For the i-th training point $x^{(i)}$, we call the "gap", geometric margin $\gamma^{(i)}$, which is the distance to the decision hyperplane and can be found by considering a point $\mathbf{x}$ on it (note that $\mathbf{w}^T\mathbf{x} = -b$):

$$\begin{aligned}\gamma^{(i)} &= \left|\frac{\mathbf{w}^T}{\|\mathbf{w}\|}\left(\mathbf{x}^{(i)} - \mathbf{x}\right)\right| \\ &= \frac{1}{\|\mathbf{w}\|} y^{(i)}\left(\mathbf{w}^T\mathbf{x}^{(i)} + b\right)\end{aligned}$$

However, we only consider the closest point $\mathbf{x}^{(i^*)} : i^* = \arg\min_i \gamma^{(i)}$ with the geometric margin of $\gamma = \min_i \gamma^{(i)}$. Since scaling $\mathbf{w}$ and b does not change the output of the classifier, nor the geometric margin $\gamma^{(i)}$, for convenience, we decide to scale $\mathbf{w}$ and b such that $\left|\mathbf{w}^T\mathbf{x}^{(i^*)} + b\right| = y^{(i^*)}\left(\mathbf{w}^T\mathbf{x}^{(i^*)} + b\right) = 1$. Thus, maximising the geometric margin of the closest point becomes

$$\begin{aligned}\max_{\mathbf{w},b} \quad & \frac{1}{\|\mathbf{w}\|} \\ \text{s.t.} \quad & y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, \quad i = 1, \ldots, m.\end{aligned}$$

which is equivalent to

$$\begin{aligned}\min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y^{(i)}(\mathbf{w}^T\mathbf{x}^{(i)} + b) \geq 1, \quad i = 1, \ldots, m.\end{aligned} \tag{5.5}$$

The solution to the optimisation problem (5.5), which can be found using off-the-shelf Quadratic Programming (QP) packages, gives us the parameters required to do classification.

Lagrange duality

While solving the optimisation problem (5.5) using QP solves the problem, we move on to derive the dual form of the problem using the principle of Lagrange duality, which will help us to solve the problem using a

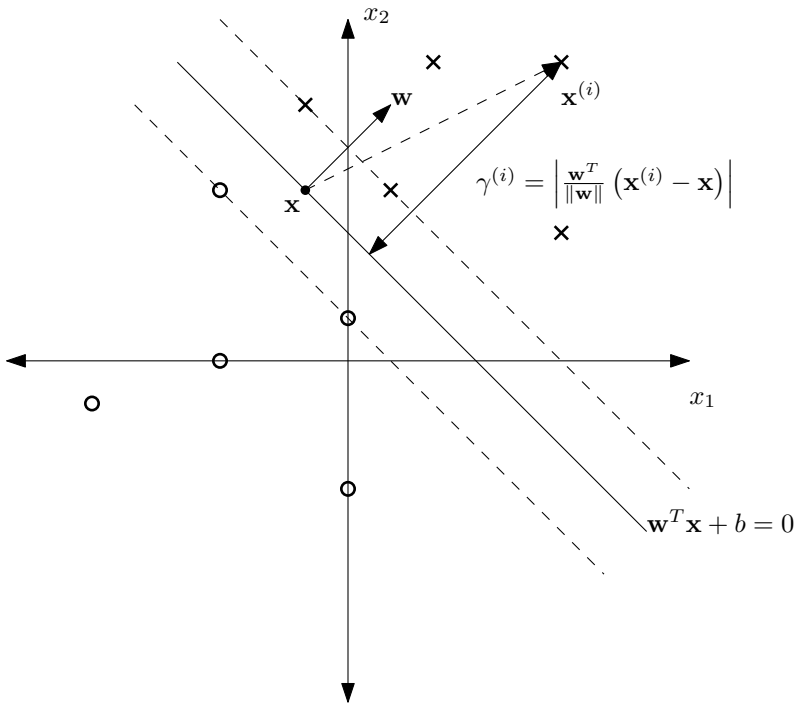


Figure 5.13: Illustration of the optimal margin classifier.

more efficient algorithm. More importantly, it will also allow us to efficiently transform the data points to high-dimensional spaces using the "kernel trick", capturing the non-linear nature of the decision boundary without losing the generalisation guarantees. The Lagrangian of our primal minimisation problem (5.5) (with parameters $\boldsymbol{\alpha} \in \mathbb{R}^m$) can be formed as

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left[y^{(i)} \left(\mathbf{w}^T \mathbf{x}^{(i)} + b \right) - 1 \right] \tag{5.6}$$

To find the dual form of the problem, $W(\boldsymbol{\alpha}) = \min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha})$, we find the gradients of (5.6) with respect to $\mathbf{w}$ and b and set them to zero to get

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)} \tag{5.7}$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \tag{5.8}$$

Substituting (5.7) and (5.8) back to (5.6) gives us the dual form, $W(\boldsymbol{\alpha})$

$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle \tag{5.9}$$

Thus, under certain conditions which are in this case fulfilled (and omitted for clarity), our original optimisation problem (5.5) becomes equivalent to the dual optimisation problem

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & W(\boldsymbol{\alpha}) \\ \text{s.t.} \quad & \alpha_i \geq 0, \qquad\qquad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0, \quad i = 1, \ldots, m \end{aligned} \tag{5.10}$$

This problem can be solved using a QP algorithm, however a more efficient Sequential Minimal Optimization (SMO) algorithm can be used. Once the optimal value $\boldsymbol{\alpha}^*$ is obtained, we can find the corresponding parameters of the SVM $\mathbf{w}^*$ using (5.7) and b^* using

$$\begin{aligned} b^* &= -\frac{\max_{i:y^{(i)}=-1} \mathbf{w}^{*T}\mathbf{x}^{(i)} + \min_{i:y^{(i)}=1} \mathbf{w}^{*T}\mathbf{x}^{(i)}}{2} \\ &= -\frac{\max_{i:y^{(i)}=-1} \sum_{j=1}^{m} \alpha_i^* y^{(i)} \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle + \min_{i:y^{(i)}=1} \sum_{j=1}^{m} \alpha_i^* y^{(i)} \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle}{2} \end{aligned} \tag{5.11}$$

Thus the classification of a test point $\mathbf{x}$ can be done by evaluating the argument of $g(\mathbf{w}^{*T}\mathbf{x} + b^*)$

$$\mathbf{w}^{*T}\mathbf{x} + b^* = \sum_{i=1}^{m} \alpha_i^* y^{(i)} \left\langle \mathbf{x}^{(i)}, \mathbf{x} \right\rangle + b^* \tag{5.12}$$

We note that due to certain conditions (Karush-Kuhn-Tucker conditions) the α_i^*'s that are non-zero correspond to the $\mathbf{x}^{(i)}$'s that lie on the margin. We call these points the Support Vectors (SVs).

Kernel trick

Note that according to (5.12) (and (5.11)), we only need to evaluate the inner products of $\mathbf{x}$ and the support vectors in order to classify the point $\mathbf{x}$. This fact is used in the "kernel trick", in which a kernel function $K(\mathbf{x}, \mathbf{z})$ is used as a proxy for the inner product $\langle \mathbf{x}, \mathbf{z} \rangle$. This effectively simulates transforming the data points to another, possibly unknown, space via a transformation function $\boldsymbol{\phi}(\cdot) : \mathbb{R}^d \to \mathcal{Z}$ as long as there exists such $\mathcal{Z}$, i.e. there exists $\boldsymbol{\phi}(\cdot)$ such that $K(\mathbf{x}, \mathbf{z}) = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\phi}(\mathbf{z})$. It can be shown that a kernel $K(\cdot, \cdot)$ is valid if and only if it satisfies certain conditions called the Mercer's condtitions. This allows us to transform our input domain into much higher dimensional domains without actually doing

so which turns out to improve the time complexity significantly.

$$K_{\text{poly}}(\mathbf{x}, \mathbf{z}) = \left(a\mathbf{x}^T\mathbf{z} + c\right)^k \tag{5.13}$$

$$K_{\text{rbf}}(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma^2}\right) \tag{5.14}$$

In our application, we are going to use the kernels above (5.13, 5.14). The first one, (5.13), is the polynomial kernel which effectively transforms the input domain space into a feature space whose features are products all possible permmutations of the input features of degrees up to k. The second one, (5.14), is a radial basis kernel (RBF), whose corresponding transformation function transforms the input domain space to an infinite dimensional one. This kernel gives a large inner product to two points if they are close to each other.

Soft margin SVMs

Soft margin SVMs are a slight modification of the original problem that allows small violations of the margin. The purpose is to guarantee that the algorithm doesn't fail because of the non-separability of the input data even after using the kernel trick. In practice, these two techniques are used simultaneously. It turns out that only minor changes arise and the SMO algorithm can still be used.

Our problem

In our problem, we are going to use the `SVM` package in `MATLAB`® in order to train our model and classify sleep apnoea. The package includes implementations of both QP and SMO algorithms, as well as various kernels, including the polynomial kernel and the RBF kernel. Recall that in our problem, we have the signal $\{s_i\}_{i=1}^{T}$ and the classifiers for every K samples $\{y_i\}_{i=1}^{T/K}$, where $y_i \in \{0, 1\}$ classifies the time period represented by $\{s_j\}_{j=(i-1)K+1}^{iK}$. When using SVMs, the input vectors and the classifiers become

$$\mathbf{x}^{(i)} = \left[s_{(i-1)K+1}, \ldots, s_{iK}\right]^T$$

$$y^{(i)} = \begin{cases} 1 & \text{if } y_i = 1 \\ -1 & \text{otherwise.} \end{cases}$$

5.6.2 State-Space Models

Here, we present the State-Space Models (SSMs) (also known as Linear Dynamical Systems) based on Kevin Murphy's book [53, Chapter 18] and Prof. Zoubin Ghahramani's paper [36]. SSMs allow us to model systems that are dynamic. Unlike the SVMs, SSMs will model dependencies of the current state on the previous ones.

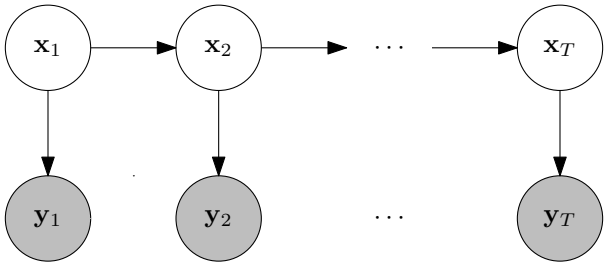


Figure 5.14: Probabilistic Graphical Model of a SSM and a HMM

We have two set of random variables, the hidden states $\mathbf{X} = \left\{\mathbf{x}_t, \mathbf{x}_t \in \mathbb{R}^d\right\}_{t=1}^T$ and the observed vectors $\mathbf{Y} = \left\{\mathbf{y}_t, \mathbf{y}_t \in \mathbb{R}^k\right\}_{t=1}^T$. We can represent SSMs graphically in a Bayesian netowrk in Figure 5.14, and write out the joint probability of the model as

$$p\left(\mathbf{X}, \mathbf{Y}\right) = p(\mathbf{x}_1)p(\mathbf{y}_1|\mathbf{x}_1)\prod_{t=2}^{T} p(\mathbf{x}_t|\mathbf{x}_{t-1})p(\mathbf{y}_t|\mathbf{x}_t) \tag{5.15}$$

In our case, we consider these particular transition and observation models with zero-mean Gaussian noises:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t \tag{5.16}$$

$$\mathbf{y}_t = \mathbf{C}\mathbf{x}_t + \mathbf{v}_t \tag{5.17}$$

$$\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \tag{5.18}$$

$$\mathbf{v}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \tag{5.19}$$

Since we assume that $\pi = p(\mathbf{x}_1)$ is also Gaussian, all our conditional probability distributions will be Gaussian due to the linearity of the transition and observation models. Once the parameters are obtained, the problem of inference and state estimation consists of

1. **Filtering.** We want to find $p(\mathbf{x}_t|\{\mathbf{y}_t\}_1^t)$. This is solved by the Kalman filtering algorithm.
2. **Smoothing.** We want to find $p(\mathbf{x}_t|\mathbf{Y})$. This is solved by the Kalman smoothing algorithm.

3. **Prediction.** We want to find $p(\mathbf{x}_{t+\tau}|\{\mathbf{y}_t\}_1^t)$. This is done by solving to 1. and then simulating using the state transition function.

We are mainly interested in filtering and smoothing.

Our problem

Although SSMs are well-suited for time-series data, they are not very well suited for our problem because the hidden state $\mathbf{x}_t$ in this model is continuous, whereas in our problem, the sleeper's state is binary. While there are techniques to map from a continuous domain to a discrete one, we move on to discuss a more promising model.

5.6.3 Hidden Markov Models

Here, we present the Hidden Markov Models (HMMs), one of the most popular statistical models in machine learning with applications in many fields including but not limited to Cryptanalysis, Speech recognition and Bioinformatics [97]. The material presented here is based on [71], [73] and [53]. We will first present HMMs with discrete observations, then we extend this to include models with continuous observations.

Discrete observations

Similarly to SSMs, we have two sets of random variables. Observed variables $\mathbf{Y} = \{y_t\}_{t=1}^T$ which are drawn from the observation alphabet $V = \{v_1, \ldots, v_M\}$, and hidden states $\mathbf{X} = \{x_t\}_{t=1}^T$ which are drawn from the hidden state alphabet $S = \{s_1, \ldots, s_N\}$. The probabilistic graphical model of the HMM is identical to the one in Figure 5.14. The hidden states obey Markov assumptions, i.e. $P(x_t|x_{t-1}, \ldots, x_1) = P(x_t|x_{t-1}) = \text{const.}, t = 2, \ldots, T$. Moreover, the observed variables are only dependent on the corresponding hidden state, i.e. $P(y_t|x_t, \ldots, x_1, y_{t-1}, \ldots, y_1) = P(y_t|x_t), t = 1, \ldots, T$.

We parameterise a HMM using the transition matrix $\mathbf{A}$, emission matrix $\mathbf{B}$, and the initial state distribution $\boldsymbol{\pi}$. The transition matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ describes the transitions between hidden states, $A_{ij} = P(x_{t+1} = s_j | x_t = s_i)$. The emission matrix $\mathbf{B} \in \mathbb{R}^{N \times M}$ describes the probability of an observation conditioned on a hidden state, $B_{jk} = B_j(v_k) = P(y_t = v_k | x_t = s_j)$. The initial state distribution $\boldsymbol{\pi} \in [0, 1]^N$ simply describes the initial probabilities of the hidden state, $\pi_i = P(x_1 = s_i)$. The model is fully described if we know these parameters, which we group into what is called a parameter set of the model, $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$.

The three main questions of a HMM are

1. Find the probability of observations given the model, $P(\mathbf{Y}; \lambda)$.
2. Find the most likely series of hidden states $\mathbf{X}$ to have generated the observations $\mathbf{Y}$, $\mathbf{X}^* = \arg\max_{\mathbf{X}} P(\mathbf{Y}|\mathbf{X}; \lambda)$.
3. Find the parameters λ to maximise $P(\mathbf{Y}; \lambda)$.

We will discuss algorithmic solution to these three problems in turn.

Solution to the first problem. To find the probability of an observed sequence, we use the dynamic programming algorithm, called the FORWARD PROCEDURE (outlined in Algorithm 1) which calculates the forward variable $\alpha_t(i) = P(\mathbf{Y}, x_t = s_i; \lambda)$. As we can see, the algorithm has a time complexity of $O(TN)$.

Algorithm 1 FORWARD PROCEDURE for computing $\alpha_t(i)$.

1. **Initialisation.**
$$\alpha_1(i) = \pi_i B_i(y_1), 1 \leq i \leq N$$
2. **Induction.**
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) A_{ij}\right] B_j(y_{t+1}), \quad \begin{matrix} 1 \leq t \leq T-1 \\ 1 \leq j \leq N \end{matrix}$$
3. **Termination.** (solution to the first problem)
$$P(\mathbf{Y}; \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Solution to the second problem. To solve the problem of finding the most likely sequence of hidden states, we use the VITERBI ALGORITHM proposed by Andrew Viterbi in 1967. The most likely sequence of hidden states is also called the Viterbi path. Firstly, we define the quantity $\delta_t(i)$, which stores the highest probability along a single path ending at the state $x_t = s_i$:

$$\delta_t(i) = \max_{x_1, \ldots, x_{t-1}} P(x_1, \ldots, x_{t-1}, x_t = s_i, y_1, \ldots, y_t; \lambda) \tag{5.20}$$

By induction, we have

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) A_{ij}\right] B_j(y_{t+1}) \tag{5.21}$$

We also keep track of the index of the hidden state that maximises this quantity in

$$\psi_{t+1}(j) = \arg\max_i \delta_t(i) A_{ij} \tag{5.22}$$

Having defined these quantities, we present the VITERBI ALGORITHM in Algorithm 2. As we can see, the

Algorithm 2 VITERBI ALGORITHM for computing the most likely sequence of hidden states.

1. **Initialisation.**

$$\begin{aligned} \delta_1(i) &= \pi_i B_i(y_1), && 1 \leq i \leq N \\ \psi_1(i) &= 0, && 1 \leq i \leq N \end{aligned}$$

2. **Recursion.**

$$\begin{aligned} \delta_t(j) &= \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) A_{ij}\right] B_j(y_t), && \begin{matrix} 2 \leq t \leq T \\ 1 \leq j \leq N \end{matrix} \\ \psi_t(j) &= \arg\max_{1 \leq i \leq N} \left[\delta_{t-1}(i) A_{ij}\right], && \begin{matrix} 2 \leq t \leq T \\ 1 \leq j \leq N \end{matrix} \end{aligned}$$

3. **Termination.**

$$\begin{aligned} P^* &= \max_{1 \leq i \leq N} \delta_T(i) \\ x_T^* &= \arg\max_{1 \leq i \leq N} \delta_T(i) \end{aligned}$$

4. **Path backtracking.**

$$x_t^* = \psi_{t+1}\left(x_{t+1}^*\right), T - 1 \geq t \geq 1$$

5. **Return** $\{x_t^*\}_1^T$.

algorithm has a time complexity of $O(TN^2)$.

Solution to the third problem. To learn the parameters of the model, $\lambda = (\mathbf{A}, \mathbf{B}, \boldsymbol{\pi})$, we use the FORWARD-BACKWARD ALGORITHM by Baum-Welch. We will need to introduce few quantities. Firstly, we introduce the backward variable $\beta_t(i) = P\left(y_{t+1}, \ldots, y_T | x_t = s_i; \lambda\right)$ which can be computed using a dynamic programming algorithm in Algorithm 3. We also define the variable $\gamma_t(i) = P\left(x_t = s_i | \mathbf{Y}; \lambda\right)$

Algorithm 3 BACKWARD ALGORITHM for computing $\beta_t(i)$.

1. **Initialisation.**
$$\beta_T(i) = 1, 1 \leq i \leq N$$
2. **Induction.**
$$\beta_t(i) = \sum_{j=1}^{N} A_{ij} B_j(y_{t+1}) \beta_{t+1}(j), \begin{array}{l} T-1 \leq t \leq 1 \\ 1 \leq j \leq N \end{array}$$

which can be expressed as

$$\begin{aligned} \gamma_t(i) &= P\left(x_t = s_i | \mathbf{Y}; \lambda\right) \\ &= \frac{P\left(x_t = s_i, \{y_\tau\}_1^t; \lambda\right) P\left(\{y_\tau\}_{t+1}^T | x_t = s_i; \lambda\right)}{P\left(\mathbf{Y}; \lambda\right)} \\ &= \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \end{aligned} \tag{5.23}$$

We also define the quantity $\xi_t(i,j)$ as

$$\begin{aligned} \xi_t(i,j) &= P\left(x_t = s_i, x_{t+1} = s_j | \mathbf{Y}; \lambda\right) \\ &= \frac{\alpha_t(i) A_{ij} B_j(y_{t+1}) \beta_{t+1}(j)}{\sum_{i,j=1}^N \alpha_t(i) A_{ij} B_j(y_{t+1}) \beta_{t+1}(j)} \end{aligned} \tag{5.24}$$

Now, we are in a position to present the FORWARD-BACKWARD ALGORITHM by Baum-Welch in Algorithm 4. The algorithm belongs to a family of Expectation-maximisation (EM) algorithms for finding maximum likelihood (ML) or maximum a posteriori (MAP) estimates of parameters in statistical models [96].

Algorithm 4 FORWARD-BACKWARD ALGORITHM (BAUM-WELCH) for estimating HMM parameters λ.

1. **Initialisation.** Set $\mathbf{A}$, $\mathbf{B}$, $\boldsymbol{\pi}$ to be random valid probability matrices/vectors.
2. **Repeat until convergence:**
   - **E-step.** Run FORWARD and BACKWARD PROCEDURES to get $\alpha_t(i)$ and $\beta_t(i)$. Evaluate $\gamma_t(i)$ using (5.23).
   - **M-step.** Re-estimate parameters using

$$\begin{aligned} \pi_i &= \gamma_1(i) \\ A_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)} \\ B_j(v_k) &= \frac{\sum_{t=1, \text{s.t.} y_t = v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \end{aligned}$$

Parameters estimation with known hidden states. In case the hidden states $\mathbf{X}$ are given to us, in addition to the observed variables $\mathbf{Y}$, the problem of parameters estimation simply reduces to counting transitions, i.e.

$$A_{ij} = \frac{\sum_{t=1}^{T-1} 1\{x_t = s_i \wedge x_{t+1} = s_j\}}{\sum_{t=1}^{T} 1\{x_t = s_i\}} \tag{5.25}$$

$$B_j(k) = \frac{\sum_{t=1}^{T} 1\{x_t = s_j \wedge y_t = v_k\}}{\sum_{t=1}^{T} 1\{x_t = s_j\}} \tag{5.26}$$

where $1(\cdot)$ is an indicator function (1 if the boolean argument is true, 0 otherwise).

Continuous observations

We now discuss the case in which the observations $\mathbf{Y} = \{y_t\}_1^T$ are not drawn from a finite set V, but are real-valued vectors $\{\mathbf{y}_t\}_1^T$, drawn from $\mathbb{R}^d$, i.e. $\mathbf{y}_t \in \mathbb{R}^d, 1 \leq t \leq T$ (Note: hidden states $\mathbf{X} = \{x_t\}_1^T$ are still drawn from a finite set S). In this case, we cannot describe the observations using an emission matrix anymore. Since the observation random variables are now continuous, they must be drawn from some probability distribution function (pdf). This function can be any arbitrary pdf, however in order to learn anything useful, we parameterise it. We assume a simple case and let it be a Gaussian, parameterised by its mean, and covariance. In particular, $B_j(\cdot)$ becomes a probability distribution function:

$$\begin{aligned} B_j(\mathbf{y}_t) &= P\left(\mathbf{y}_t | x_t = s_j\right) \\ &= \mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_j, \mathbf{U}_j), \qquad \begin{matrix} 1 \leq t \leq T \\ 1 \leq j \leq N \end{matrix} \end{aligned} \tag{5.27}$$

This means that instead of the original emission matrix $\mathbf{B}$, we are now parameterising the emissions using the means $\boldsymbol{\mu} = \{\boldsymbol{\mu}_j \in \mathbb{R}^d\}_1^N$ and the covariances $\mathbf{U} = \{\mathbf{U}_j \in \mathbb{R}^{d \times d}\}_1^N$. Thus, our parameters set becomes $\lambda = \{\mathbf{A}, \boldsymbol{\mu}, \mathbf{U}, \boldsymbol{\pi}\}$.

Changes to the algorithms. While the three main questions for the HMM remain the same, we must change the algorithms used in the discrete observations case. It turns out that in the Forward Procedure (Algorithm 1), Backward Procedure (Algorithm 3), and the Viterbi Algorithm (Algorithm 2), we don't need to change anything except the interpretation of $B_j(\cdot)$. Whereas before, we had a single value for this quantity, now we must evaluate it using the parameters of the pdf (in this case mean and covariance) in (5.27). The problem of estimating parameters, given the observations becomes slightly

different and will be left untouched in our report. However, we discuss parameters estimation, given both the observations and the hidden states, which simply becomes fitting a Gaussian, i.e.

$$\boldsymbol{\mu}_j = \frac{\sum_{t=1}^{T} 1\{x_t = s_j\}\mathbf{y}_t}{\sum_{t=1}^{T} 1\{x_t = s_j\}} \tag{5.28}$$

$$\mathbf{U}_j = \frac{\sum_{t=1}^{T} 1\{x_t = s_j\}(\mathbf{y}_t - \boldsymbol{\mu}_j)(\mathbf{y}_t - \boldsymbol{\mu}_j)^T}{\sum_{t=1}^{T} 1\{x_t = s_j\}} \tag{5.29}$$

Different probability density functions. We have assumed that the conditional distribution of the observations is Gaussian. One disadvantage of this method is that it may be too simple to capture the real pdf the observations are drawn from. One of the alternatives is the Gaussian Mixture Model (GMM), however fine-tuning the number of modes can be difficult. It turns out that finding the optimal MLE for a GMM is intractable [53].

Our problem

In our problem, we model the apnoeatic states as the hidden states $\{x_t\}_1^T$ drawn from a binary set $\{0, 1\}$. Although the observed signal is a sampled one-dimensional signal, since we only have annotations every K samples, we stack all K samples to a vector and treat it as a K-dimensional, real-valued observed variable $\mathbf{y}_t \in \mathbb{R}^d$, $(d = K)$. Since we assume that we have the annotated signal, we can do the training offline. Thus we are interested in the third problem (with known hidden states), for which the solution is just fitting the parameters; and the second problem, finding the most likely sequence of the apnoeatic diagnoses, given the model and the observations, using the VITERBI ALGORITHM. We will train the model using the annotated data and once trained, we will use the trained model to diagnose sleep apnoea from new data. Implementations of the algorithms for our applications are available for `MATLAB®` in the packages `pmtk3` (`https://github.com/probml/pmtk3`) and `HMM Toolbox` (`http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`) from Kevin Murphy.

5.6.4 Conditioning input data

Working in the time domain, the data is usually transformed to the frequency domain to analyse where the pattern is found more easily. Also, ergardless of the chosen model, we will work in high dimensions which means we need a lot of input data in order to find the pattern. In the case of SVMs, it will be high-dimensional training points $\{\mathbf{x}^{(i)}\}_1^m$; in the case of SSMs and HMMs, it will be the high-dimensional observed variables $\{\mathbf{y}_t\}_1^T$. We will present two ways to condition the input data, before analysing it using

the learning algorithms above, namely frequency-domain analysis and Principal Components Analysis.

Frequency-domain analysis

Discrete Fourier Transform. Discrete Fourier Transform (DFT) is used to transform a finite list of equally spaced samples of a function into the list of coefficients of a finite combination of complex sinusoids [95]. It can be said to convert the sampled function from its original domain to the frequency domain [95]. The DFT, $\mathcal{F}$, can be defined to transform $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$ to $\mathbf{X} = \{X_1, X_2, \ldots, X_N\}$ as

$$X_k = \sum_{n=1}^{N} x_n \exp\left(-i2\pi kn/N\right), 1 \leq k \leq N \tag{5.30}$$

and can be performed using the Fast Fourier Transform (FFT) algorithm.

Power Spectral Density. It is common to take the Power Spectral Density (PSD) instead of the DFT in order to remove the imaginary components in the frequency domain. The PSD of a continuous signal $x(t)$ can be defined as

$$S_{xx}(f) = \lim_{T\to\infty} \frac{1}{T} |X(f)|^2 \tag{5.31}$$

where $X(f)$ is the Fourier transform ($\mathcal{FT}$) of $x(t)$ in the interval $-T/2 < t < T/2$ [38]. It can be shown that in the discrete case where $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$, the PSD is obtained by

$$S_{xx}(k) = \lim_{T\to\infty} \frac{(\Delta t)^2}{T} |X_k|^2 \tag{5.32}$$

$$\approx \frac{(\Delta t)^2}{T} |X_k|^2 \tag{5.33}$$

where T is the actual recording time, N is number of samples and $1/\Delta t$ is the sampling frequency (note that $T = N(\Delta t)$) [99].

Spectrograms. To create a spectrogram of a signal $\mathbf{x} = \{x_1, x_2, \ldots, x_N\} = \{x(\Delta t), x(2\Delta t), \ldots, x(N\Delta t)\}$, we must choose a window size T_{window} and an offset T_{offset}. Then, we keep shifting the window by T_{offset} and each time, we transform and store the window's content (in time domain) to the frequency domain using the DFT. This way, we will have a frequency domain data of the size T_{window} every T_{offset}, which we will call $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_\tau$. We usually represent this frequency domain data using colour intensities and plot them as τ columns of colour intensities with time on the horizontal axis and frequency on the vertical

axis. Figure 5.15 illustrates the process of creating a spectrogram. However, it is common to find the PSDs instead of the DFTs of the sliding windows (the principle stays the same). Hence we will refer to the process of creating the spectrogram with PSDs (instead of DFTs) of the sliding windows as *freqency analysis of the signal*.

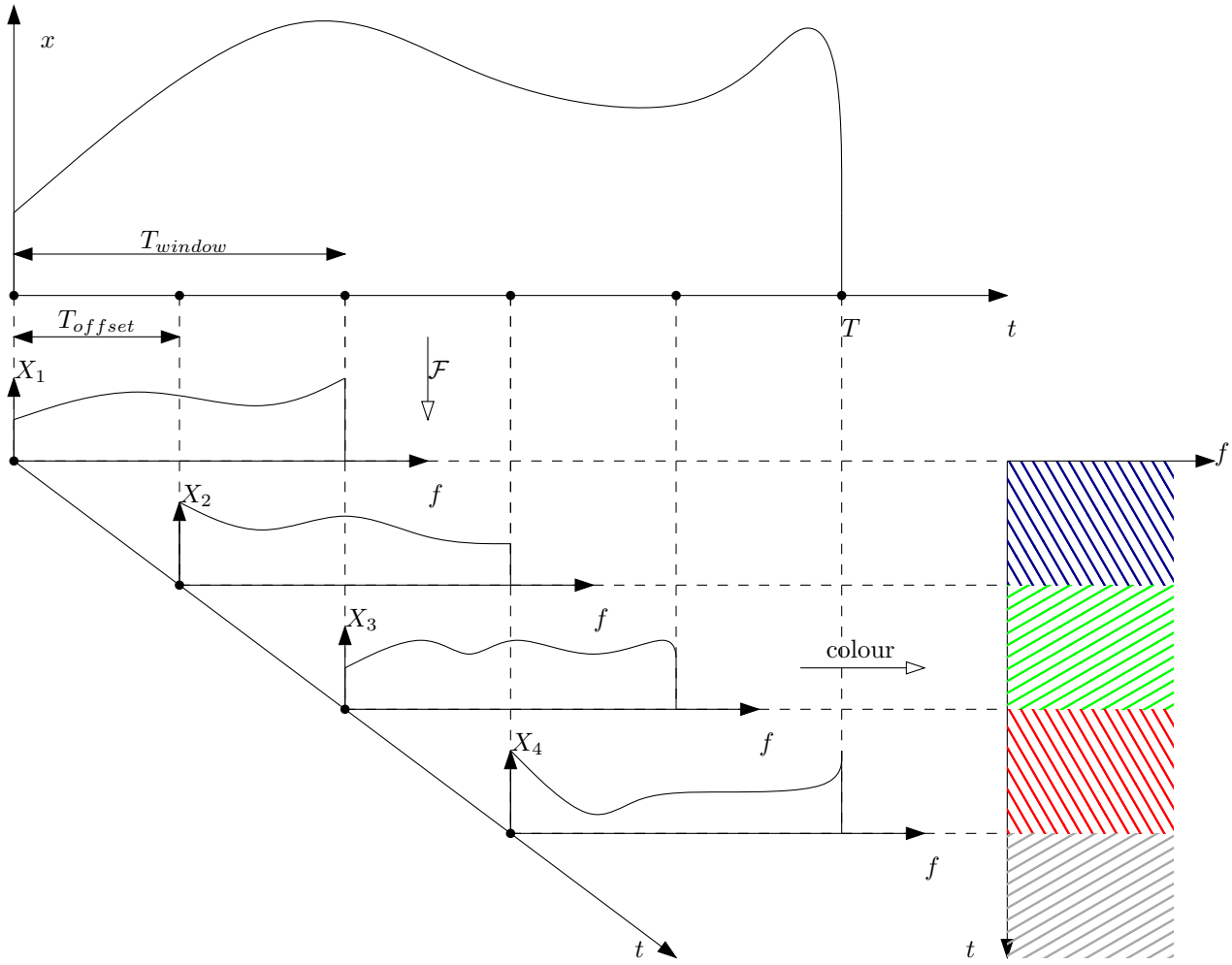


Figure 5.15: The process of creating a spectrogram.

Principal Components Analysis

Here we present the theory for Principal Components Analysis (PCA) based on Andrew Ng's lecture notes [56] and Kevin Murphy's book [53]. The objective is to take m n-dimensional input data points $\{\mathbf{x}^{(i)} \in \mathbb{R}^n\}_1^m$ and transform them into m k-dimensional data points ($k < n$) $\{\mathbf{y}^{(i)} \in \mathbb{R}^m\}_1^m$, where $\mathbf{y}^{(i)}$'s are projections of $\mathbf{x}^{(i)}$'s onto k orthonormal basis vectors $\{\mathbf{u}_i\}_1^k$ while "preserving the most variance". We assume that over all m points, their mean $\left[\{x_j^{(i)}\}_{i=1}^m\right] = 0$ and their var $\left[\{x_j^{(i)}\}_{i=1}^m\right] = 1$ for all features of the input points $1 \leq j \leq n$. We normalise the data first if this is not the case. For $n = 2$, $k = 1$, Figure

5.16 shows the projections to the new axis.

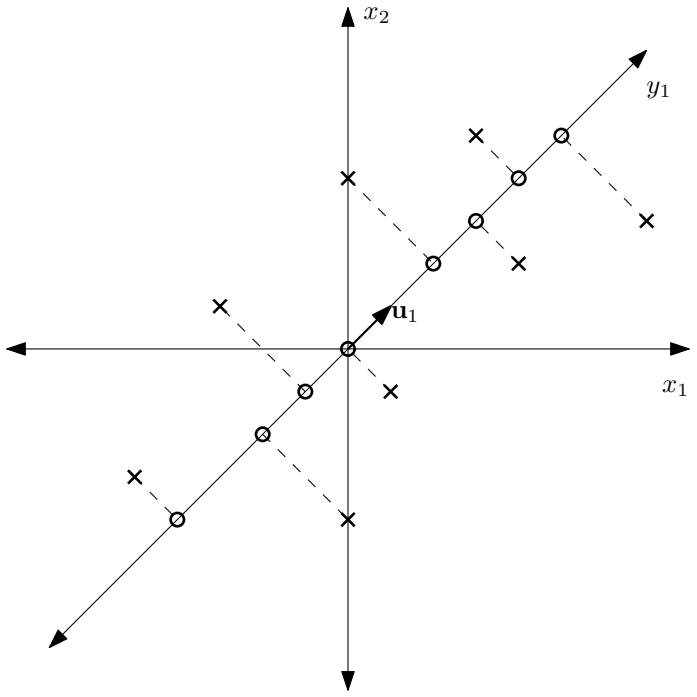


Figure 5.16: Illustration of PCA.

We can see that the transformed points can be expressed as

$$\mathbf{y}^{(i)} = \begin{bmatrix} \mathbf{u}_1^T \mathbf{x}^{(i)} \\ \mathbf{u}_2^T \mathbf{x}^{(i)} \\ \vdots \\ \mathbf{u}_k^T \mathbf{x}^{(i)} \end{bmatrix}, 1 \leq i \leq m \tag{5.34}$$

It can be shown that in order to maximise $\sum_{i=1}^{m} \left\| \mathbf{y}^{(i)} \right\|^2$, conditioned on orthonormality of the bases, we must choose normalised eigenvectors corresponding to the k largest eigenvalues of the variance matrix $\mathbf{\Sigma} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{x}^{(i)} {\mathbf{x}^{(i)}}^T$ as the bases. These bases are also called the *principal components*. The eigenvalues are proportional to the actual variances of data in the new bases. Hence, we can compare the relative "importance" of the bases and use this fact to decide how many principal components to use.

5.6.5 Summary

We have explored the theory behind three models: Support Vector Machines (SVMs), State-Space Models (SSMs), and Hidden Markov Models (HMMs). While the SVMs model data discriminatively, the SSMs and HMMs do it generatively. In our project, we will use SVMs and HMMs to learn from actual sleep

data and use the learnt model to classify unseen test data. We compare the performance of both methods by comparing the proportion of correct classifications.

We have also discussed two methods to condition our data: using spectrograms and Principal Components Analysis (PCA). In our experiments, we will transform our data into spectrogram space and then perform PCA to choose only a few principal components to reduce the dimensionality of our data.

The model with the best performance, together with the conditioning methods will be implemented in Java for Android integration.

5.7 Machine Learning – Experiments

We obtained the data from the ECG database from the PhysioNet database located at `http://physionet.org/physiobank/database/apnea-ecg/`. The data consists of 35 labelled training records and 35 unlabelled records (used for the CinC Challenge 2000 competition). The recordings vary from less than 7 hours to 10 hours each and include continuous digitised ECG signal and, in the case of the training data, a set of apnoea annotations derived by human experts. The continuous signal is sampled at the rate of 100 Hz and the annotations are available at every 6000 samples (i.e. every minute) of the signal indicating the presence of apnoea at that time. One such record is shown in Figure 5.17.

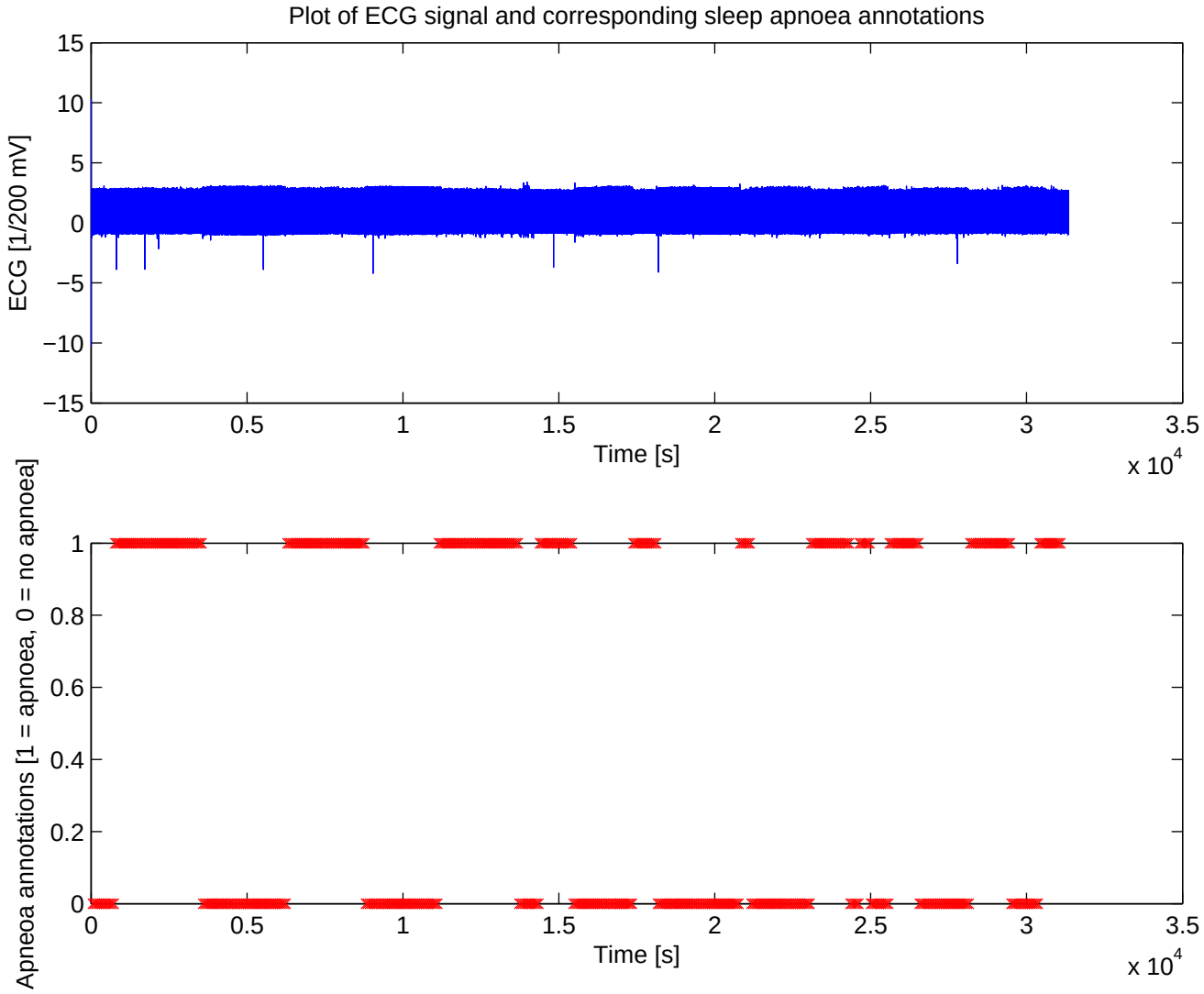


Figure 5.17: One training record

5.7.1 Reading and Conditioning Data in MATLAB

The code used for the training and testing of the data can be found in Appendix E for ease of explanation. Firstly, we examine the main script, covering the reading and conditioning of data, spectrogram transformation and PCA analysis. After explaining the various functions created and used in the script till then, we move on to cover the training and testing of the data.

From the code found in the Appendix E.1, the `trainIndex` and `testIndex` vectors are used simply for selecting the files to be used, out of 35, for training and the remainder (or less) for testing the accuracy of the diagnosis. Having chosen the files, the next step is to extract and read the files. This is done using the `readData` function, presented in Appendix B.

The function reads data from the indices specified in `fileIndex` (which is trainIndex in the main script above), and returns O, containing all the observations merged together in a TxD matrix. T is the total number of minutes of data, and D is the number of samples in a minute (6000 in this case), such that there are T annotations in total. The function also returns the vector q, a Tx1 vector containing the latent states for every minute in O, as well as consolidated time, signal and annotation time vectors for ease of plotting and analysis later on.

Firstly, `readData` uses a simple function `getFilenames` to return a 35x1 cell of the available filenames, in a cell string. Then, after initialising the variables, `readData` uses a for-loop to run through each file and extract the relevant information. Using the pre-provided `rdsamp` and `rdann` functions, the signal values as well as annotations are read from the file. As the annotations use 'A' for apnoeatic episodes and 'N' for non-apnoeatic episodes, the vector type is converted to the alphabet $0, 1$. The `O` and `q` output matrices are built up using the information from each file, and finally some trivial conditioning is done to ensure ease of plotting if the signal were to be kept.

Armed with the consolidated vectors `X` and `Y`, we now proceed to use the `spectrogram` function in `MATLAB`®, as described in section 5.7.2. We then use the `pca` function from the `pmtk3` package to perform Principal Components Analysis (choosing the number of principal components we wish to include, `k`).

5.7.2 Conditioning input data

Frequency Analysis

We use `MATLAB`®'s function `spectrogram`, which takes the following parameters

- `X` – our signal $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$.

- `WINDOW` – length (in number of samples) of the window T_{window}. The windows are automatically filtered using a Hamming window. We arbitrarily choose the window to be the number of signal samples corresponding to one annotation (6000 in our case).
- `NOVERLAP` – number of overlapping samples between two consecutive windows. Effectively $T_{\text{window}} - T_{\text{offset}}$. We arbitrarily choose the offset to be 1000 such that we have six bins of PSD's for each annotation.
- `NFFT` – number of frequency points used to calculate the discrete Fourier transforms. We use default.
- `Fs` – sampling frequency in Hz. In our case 100 Hz.

A spectrogram for one record can be seen in Figure 5.18. Just by looking at the spectrogram, we can spot different regimes of the time series (most clearly seen at 0.5×10^4, 1×10^4, 2×10^4, and 2.7×10^4 seconds). Comparing this to the apnoea annotations in Figure 5.17, these regimes are actually the non-apnoeatic regimes. Also, we can see that most of the "action" is happening below 20 Hz. For this reason, we cut off the frequencies above 25 Hz for subsequent analysis. We combine the corresponding six bins of PSD's per annotation in to a large feature vector corresponding to that annotation. The actual implementation can be found in Appendix C.1. Next, we will reduce the dimensionality of these feature vectors using PCA.

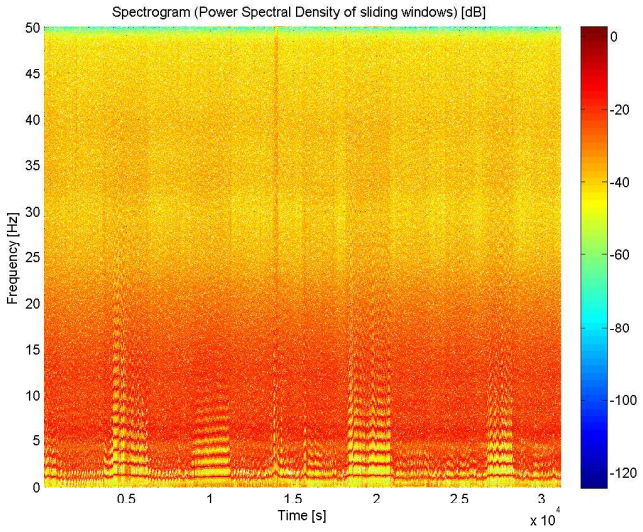


Figure 5.18: Spectrogram of one record

PCA

For this, we use the function `pca` from the package `pmtk3`. Once we get the principal components (orthogonal bases) and their corresponding principal coefficients (variances of the data, projected onto

that base, correct to a constant factor) $\{\lambda_1, \lambda_2, \ldots, \lambda_D\}$, we will plot the graph of their cumulative sum over their total sum $\frac{\sum_{i=1}^{n} \lambda_i}{\sum_{j=1}^{N} \lambda_j}$ in order to decide on the number of principal components needed. The plot is in the Figure 5.19. Performed on the first 10 records, we can see that we will need to include at least 100 to capture half of the variance but at least 3000 components to capture the whole variance. The actual implementation can be found in Appendices C.2 and C.3.

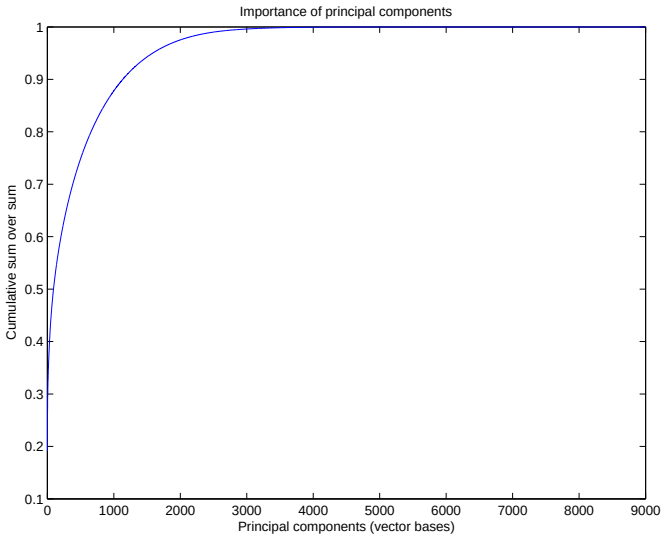


Figure 5.19: PCA of the first 10 records

5.7.3 Data Visualisation

It is helpful to visualise our data to confirm the presence of any detectable patterns and/or to help improve our learning algorithms. Since our Principal Component Analysis revealed that we need at least 100 principal components to capture at least half of the variance, it is not illustrative to visualise our data by plotting the data points on a 2D plane using only the first two principal components. Instead, we take the six corresponding bins of spectrograms of the two classes of annotations (apnoea, no apnoea) and take the average for each class as shown in Figure 5.20. We can notice the differences in the spectrograms which is reassuring because we can see that there exist some patterns to be found and that the spectrogram contains enough information to detect sleep apnoea.

5.7.4 Support Vector Machines

Firstly, we investigate the case when no kernels are used. Then we will investigate the use of two possible kernels – polynomial (degree 3) and radial basis function. We will train on the first ten records using the `MATLAB`®'s function `svmtrain`. Then we will test our trained parameters on the next five records,

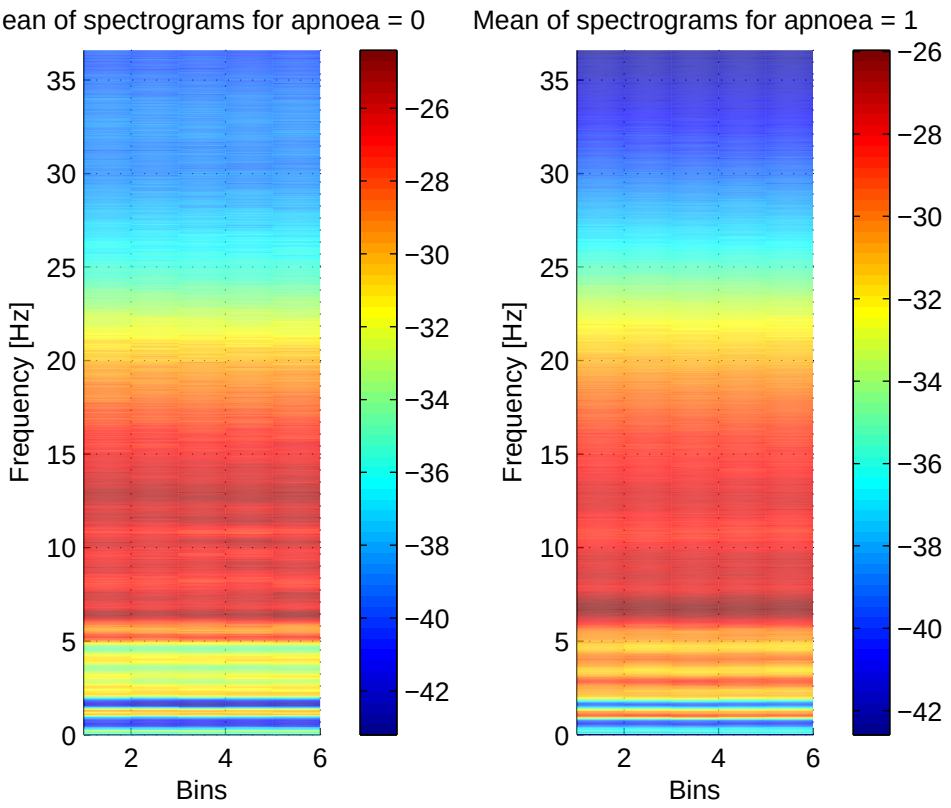


Figure 5.20: Comparison of the spectrograms of the two classes based on their average spectrogram from the first five records.

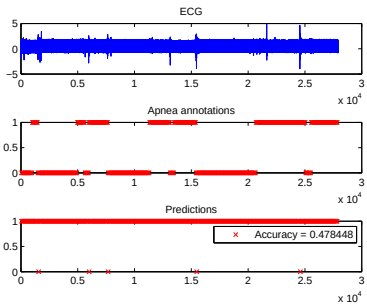
where we record the accuracy as the ratio of the number of correctly classified annotations over the total number of annotations. We will also record the number of Support Vectors (SVs) because they indicate the generalisation performance. The actual implementation can be found in Appendix D.

No kernels

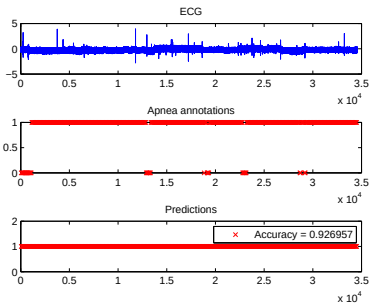
The results of this experiment are shown in Figure 5.21 and in Table 5.13. The accuracy is high, however so is the number of SV's. This indicates that the generalisation guarantee is poor.

Polynomial kernel

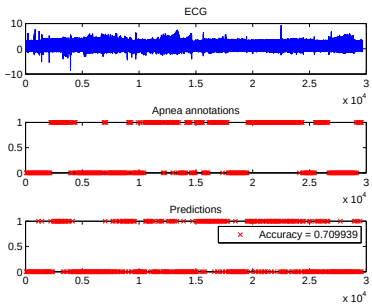
The results of this experiment are shown in Figure 5.22 and in Table 5.13. We can see that there is a low number of SVs and high accuracy. This means that the polynomial kernel is likely to perform well on unseen data. If we had more computational resources, we could have used more features (and correspondingly more training data) in order to represent the features better.



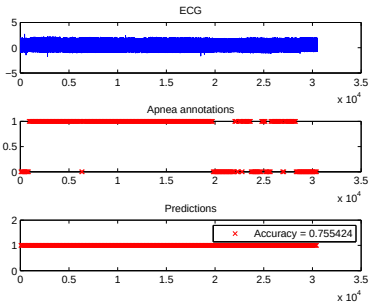
(a) record 11



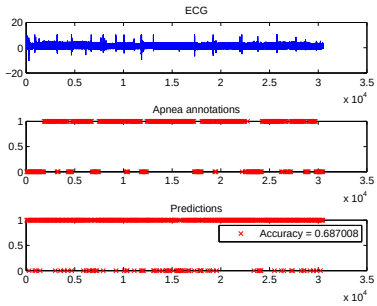
(b) record 12



(c) record 13

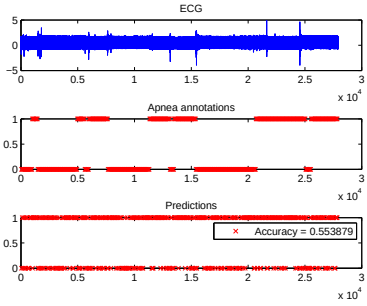


(d) record 14

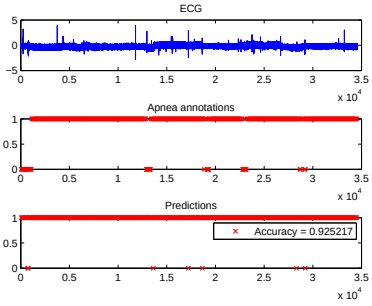


(e) record 15

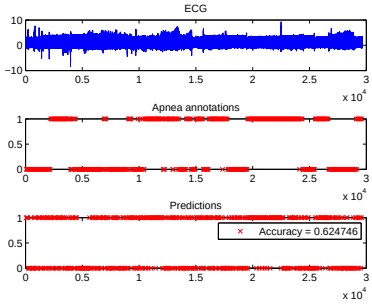
Figure 5.21: Performance of no kernels on the test records 11 to 15



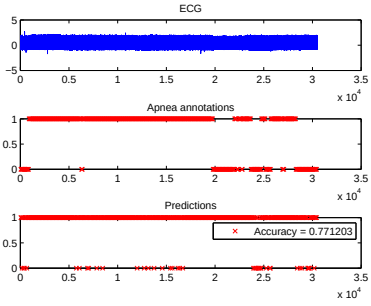
(a) record 11



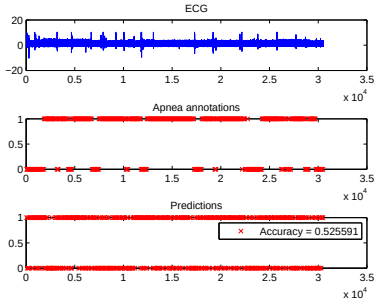
(b) record 12



(c) record 13



(d) record 14

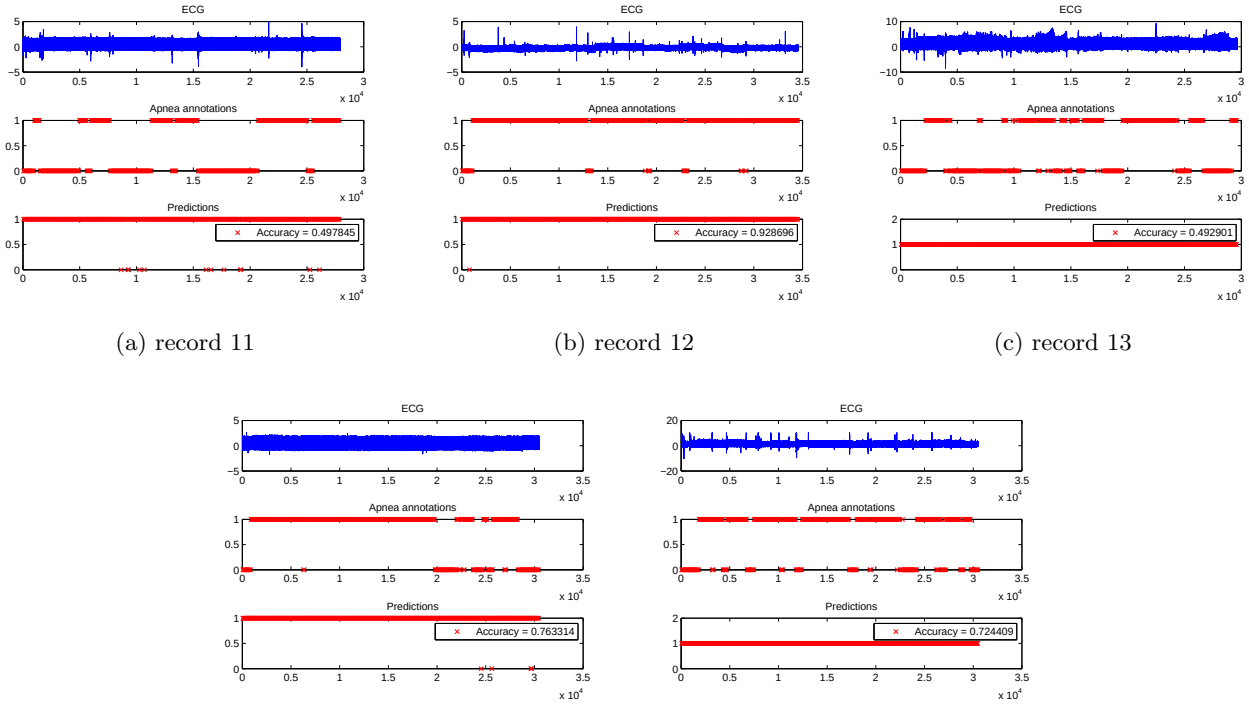


(e) record 15

Figure 5.22: Performance of the polynomial kernel on the test records 11 to 15

Radial basis function kernel

The results of this experiment are shown in Figure 5.23 and in Table 5.13. Similarly to the case of no kernels, the accuracy is high, but so is the number of SV's indicating poor generalisation performance.



(a) record 11 (b) record 12 (c) record 13

(d) record 14 (e) record 15

Figure 5.23: Performance of the Radial Basis Function kernel on the test records 11 to 15

Summary

Summary of the performances of the different kernels is shown in Table 5.13. The polynomial kernel with degree 3 is the most promising in terms of generalisation. All three kernels have similar accuracy. If we had more computational resources, we could have done deeper analysis and tests, e.g. use more principal components to capture as much variance as possible and use more training data (we only used 10 out of 35).

	# SV's (5005 data points)	Average accuracy
No kernel	3679	0.7116
Poly kernel	1879	0.6801
Rbf kernel	4033	0.6814

Table 5.13: Summary of performances of different kernels

5.7.5 Hidden Markov Models

Having examined briefly the theory behind Hidden Markov Models, let us now look at how the training was done offline, and analyse some results from subsequent tests. As mentioned earlier, the apnoeatic states are modelled as the hidden states $\{x_t\}_1^T$, and are elements of the binary set $\{0, 1\}$. The observed signal is annotated every K samples (every minute in the case of the PhysioNet data), so we stack all K samples into $\mathbf{y}_t \in \mathbb{R}^d$, $(d = K)$. Using the packages `pmtk3` and `HMM Toolbox`, the algorithms were implemented in `MATLAB`®, along with the conditioning of the data using spectrogram and PCA analysis. Again, `MATLAB`® is used for convenience and the code can be easily converted to `Java` after experimentation and analysis.

The data is read and conditioned as explained in Section 5.7.1.

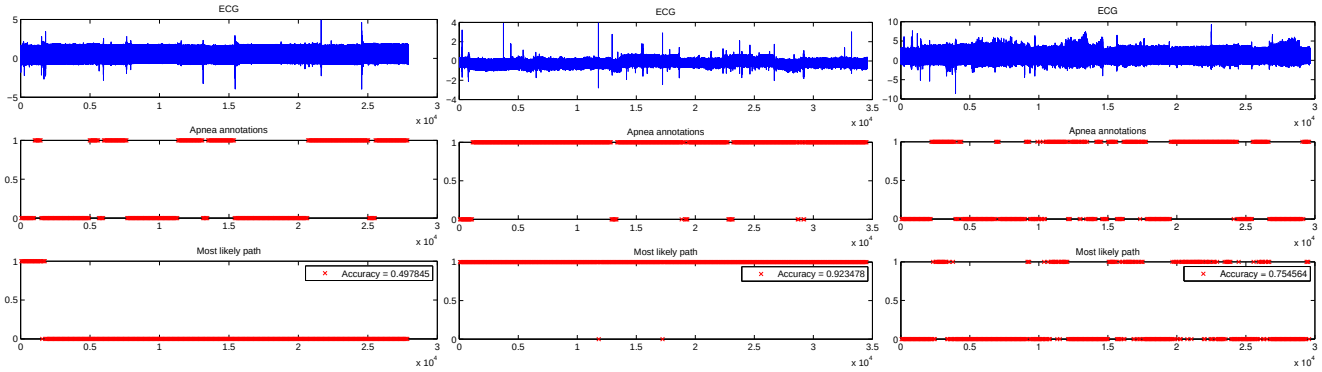
We then move on to training and fitting the parameters, using the `pmtk3` and `HMM Toolbox` packages. We then compare the expected hidden states calculated using the `Viterbi Algorithm` with the actual underlying states and determine the accuracy of the HMM model diagnosis.

The parameters are fitted using the `apneaHMMTrain` function shown in Appendix E.2, which computes the transitional matrix `A` using

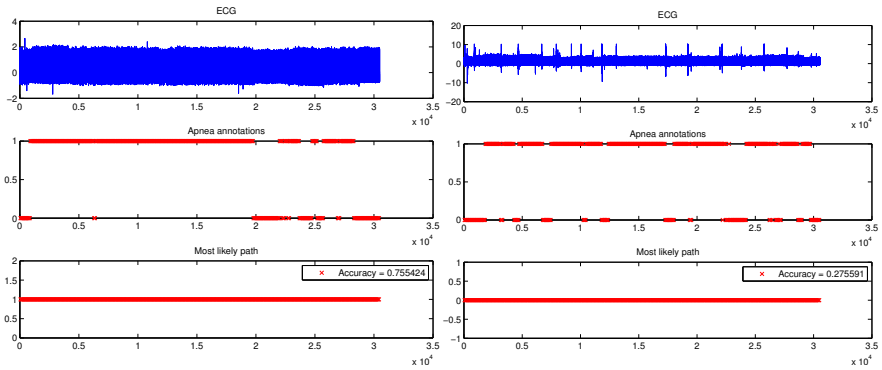
$$A_{ij} = \frac{\sum_{t=1}^{T-1} 1\{x_t = s_i \wedge x_{t+1} = s_j\}}{\sum_{t=1}^{T} 1\{x_t = s_i\}} \tag{5.35}$$

and uses Gaussian fitting to calculate the pdfs for the emissions (the `gaussFit` function is used from the `HMM Toolbox` package – we shall not go into the details on how the function is implemented here). We are left with the outputs `A`, the 2x2 transition matrix, `Mu` and `U`, parameters of the Gaussian pdf fit of the emissions, and `pi`, the initial state distribution.

In order to compare with the test data, we need to read the data specified using `testIndex`, condition it, and use the `Viterbi Algorithm` to calculate the most likely path. This is done in the function `apneaHMMTest`, found in Appendix E.3. The code for reading and conditioning the data (and plotting the results) has been ommitted as it is similar to that shown before. Once again, functions from 'off-the-shelf' packages are used in implementing the `Viterbi Algorithm` and the accuracy of the most likely path is calculated by comparing it to the annotations reflecting the true states. This is done for each test file, and the results are shown below, in Figure 5.24.



(a) record 11 (b) record 12 (c) record 13



(d) record 14 (e) record 15

Figure 5.24: Performance of HMM model on the test records 11 to 15

5.7.6 Summary

We have seen the implementation of the SVM and HMM models to the PhysioNet data and have calculated the accuracy of both methods in diagnosing apnoea from five records. While the average accuracy for the HMM model is much lower than that found for the SVM model, at 64.1%, we feel that the HMM model would still be more appropriate for use in our app. Due to limited computational power and memory (fitting the parameters takes a considerable amount of time on a standard laptop), we were unable to utilise the full set of data for our experiments. However, we are confident that if more time and effort were to be expended in this area, the HMM model would prove to be more accurate in diagnosing apnoea as compared to the SVM model. This is because of the suitability of the Hidden Markov Model for temporal data and the applicability of the Markov assumptions in our case. The accuracy can also be improved by ensuring that the annotations on training and testing data reflect the reality of the patient.

The current HMM or SVM models can be improved further as well. For example, we could set more suitable probability distributions over the observations, rather than just the multi-variate Gaussian distribution we have used (such as a mixture of Gaussians). Secondly, better dimensionality reduction

could be achieved using auto-encoders, for example.

What we have managed to prove here, however, is that diagnosis of OSA using non-invasive techinques is possible if the appropriate machine learning tools are utilised. If more resources were to be invested, an accurate and comprehensive diagnosis can be created in conjunction with the questionnaire.

5.8 Machine Learning – Java

Now that we have established the model for our problem (Hidden Markov Models), and used it to learn from data (5.7.6 Summary), we can implement it in Java for Android integration. Our implementation should be able to take sleeping data as an input (in our case observations) and output the most likely sequence of apnoea annotations (in our case latent states). Since all parameters for the HMM have been learnt in MATLAB beforehand, our implementation does not need to include the learning part. More specifically we want the input and output to be lists of integers (`List<Integer> observations` and `List<Integer> latentStates`).

For this purpose, we have created a class `ApneaHMM` with the following API:

- `void getParameters()` – reads the learnt parameters from a local file.
- `void setObservations(List<ObservationVector> observations)` – sets the observations (sleeping data) for the object.
- `void conditionData()` – conditions observations by transforming it to the spectrogram space and then performing PCA, similarly as done in the experiments (using the same principal components).
- `void infer()` – runs the Viterbi Algorithm to infer the most likely sequence of apnoea annotations given the observations based on the model parameters.
- `List<Integer> getLatentStates()` – returns the inferred most likely sequence.

For the Viterbi Algorithm, we are using the `Jahmm` library, which is a Java implementation of HMM related algorithms [33]. For the data conditioning, we are using the `Linear Algebra for Java (la4j)` library [44], and for the spectrogram analysis, we are using the `musicg`, a lightweight audio analysis library [101].

Due to time constraints and lack of audio data, we have not implemented our model in Java and leave this for future work.

5.9 Designing an Android App

In this section of the report about designing an Android app with the Android Developer Tools (ADT) software, there will not be any coding guidance on the basics of Java nor XML. As such, any examples discussed or shown will either have a short explanation or will be simple enough to assume the functionality of the code is evident. The ADT software is used in this project as it facilitates the balance in a simple to use package yet includes more detailed app functionality such as accessing the sound recorders buffer. There are many other Android app developing software packages, but none achieve this balance as well. For example the MIT app inventor provides basic building blocks for an app but limits the customisation of the code that is an essential element for our audio processing. appsgeyser.com is even simpler in providing an app interface for a pre-existing website. However we have decided not to use a web-based approach due to the extra complications from patient security and limitations in excessive data usage in streaming audio to a website.

5.9.1 XML and Java sides of ADT

The Android Developer Tools software has a simple architecture for basic apps, which uses XML code to create the user interface of buttons and graphics, and Java code to describe to the phone what each of these items in the XML code does. So for any page of the phone app, there is an XML code called the layout with an associated set of Java instructions called a Java activity. Whilst it is intuitively easier to consider a visual layout with instructions behind each section, the programming design works much more naturally the other way around, with the Java activity dictating the layout. For this particular app, we need four pages in total. Therefore we need to create a total of four Java activities to describe each page, and each one of these will call an XML layout to attach various commands to.

5.9.2 XML IDs

Each interactive item of the XML layout will have a unique ID, and the Java code will directly link an object to this, which can then be used in the functionality instructions. The following code first assigns the XML button ID btn_analyse to a Java button object called 'btnAnalyse', then informs the phone to carry out the task startAnalyseActivity() when it detects that the button has been pressed.

```
1 btnAnalyse = (Button) findViewById(R.id.btn_analyse);
```

```
2 btnAnalyse.setOnClickListener(new View.OnClickListener() {
3 public void onClick(View v) {
4 startAnalyseActivity();
5           }
6     });
```

Our application uses a minimalistic interface and as such there are few XML IDs to work with. Regardless, a convention was established such that each object is labeled by its type followed by an underscore, then its unique name (for example btn_analyse as above).

5.9.3 Methods

The task startAnalyseActivity() used in the example above is known as a method in Java code and is a set of code instructions grouped together under a single name. Not only does this make the code clearer to understand, it also allows repeated sections of code to be called much more simply: just contain all the repeated parts under one method and call that method each time. For these reasons, code for each part of the application functionality is contained in separate methods.

5.9.4 Java and android libraries

The app writer is not expected to write machine level code for the smart phone and is provided with a large array of building blocks for both Java and Android based coding which are contained in libraries which must be imported at the start of each activity. For example, the Android.media.AudioRecord library is needed to use the phones built in microphone, and it provides simple inbuilt methods such as startRecording() and stop(). Further details of this particularly important library will be discussed later.

5.9.5 Structure of an Activity

Each activity can be divided down into several key blocks that can be explained in the chronological order in which they are seen in the code:

1. Importing the necessary libraries of methods
2. Declaring the whole activity to create it
3. Declaring any variables and objects that the activity uses
4. Creating the onCreate() task upon which the activity starts running, and in which the core instructions and external methods are called

5. Defining any extra methods used, with their code detailed inside

5.9.6 XML writing

Again, it is not within the scope of this report to comprehensively explain how to code XML, but the code here is very clear to comprehend upon reading each line of code. The Android Developer Tools software allows the user to directly edit the XML code, and to also view and edit the layout from a graphical preview layout. Note that editing either one of these will update the other in real time. Each object is defined in a block enclosed by ¡ and /¿ and must contain its own ID definition as a minimum. The apps theme will dictate a large set of default values for any object parameter, so only the physical size and details unique to each object need to be added. This might include unique text for a button or a variation in font for some text all of which are edited under very clear labels such as android:fontFamily=.

5.9.7 XML layouts

The overall page layout has two main formats: linear layout and relative layout. Relative layout requires each objects relative position to be labelled in reference to at least one other object with a fixed location and allows objects to be placed alongside each other as well as above and below (effectively two dimensions of freedom). Linear layout is simpler and more constrained in that the objects will be placed in the order they appear in the XML code in one dimension. For this application, linear layout is suitable. Other useful tools here include ¡ScrollView/¿ which allows all objects within it to be viewed as a scrollable page. This was used on the questionnaire to prevent the questions becoming too small to be readable. It can also be used internally to scroll a subset of objects rather than the whole page.

5.9.8 Further XML details

When defining an objects size, its useful to use adaptable sizes to ensure good results on all devices and screen sizes (as opposed to a set height of 10px for example, which could look small on a tablet but large on a small phone). The two options here are FILL_PARENT, which makes the object as wide as the object its contained in, and WRAP_CONTENT, which makes the object as wide as necessary to contain all of its components. Using the example of a button width inside an empty android page, FILL_PARENT would make the button as wide as the screen, and WRAP_CONTENT would make the button wide enough to include all the button text.

5.9.9 Handling Strings

When creating buttons and other features with text on, the text should be stored in the separate file in res/values/strings.xml instead of hardcoding it directly into the XML code. Whilst it is less of an issue on a small-scale project such as this, it can become increasingly difficult to find where to change particular items of text in larger projects. Instead we label each part of text from the strings file and reference this ID where it is needed in the code. Then we can quickly find where to change any text in future, or perhaps reuse the same text in multiple buttons or other applications.

5.9.10 Java Techniques

Sharing data between activities can be achieved with a basic technique called SharedPreferences which stores any required variable locally in the app. As with many techniques, it requires importing its own class from the android library, and vitally requires the command editor.commit() to save the changes before finishing. It is utilised in this application to store the received information from the questionnaire. By restoring the previous stored settings upon the app initialisation, the questionnaire results are then stored on a more permanent basis (I.E when the application has been closed and reopened).

Changing from one activity to another is done using a class called Intent which is effectively setting up a requirement for something to happen. In this case the intent will provide the details of the activity to be launched. A button press might then call startActivity(intent) which will launch whatever the original intent requested. In this way it can be used as an independent launching mechanism as it is not aware of the details of the intent nor is the intent solely aimed at one target. For example, multiple buttons can launch one action, all acting as effective triggers. In this application, intents are primarily used to launch new activities.

When starting multiple activities, care needs to be taken as to how and when previous activities are closed. Whist it might seem to make sense to have the home page close upon opening a new window, this posed a subtle bug in practice. The pressing of two activity-launching buttons simultaneously causes both of them to open and this is not a significant issue. However, pressing the back button in this format requires launching the home page again. Therefore upon exiting one page, a residual page is left open behind the current home page, and upon a few repetitions of this, the app will quickly be running several pages simultaneously. Without exploring in too much depth how to make each activity-launching button press a unique event (such as a mutex-like control), the simplest solution in this case was to allow up to

three of the new activity pages to open simultaneously and leave the home page unclosed. This limits the total possible number of pages open to four. This is a very unlikely event, and is easily remedied by pressing the back button.

RadioGroup buttons are used in the questionnaire section of the app. These provide the advantages of never leaving a blank answer and being instantly comprehendible to any user. Similarly to the buttons, each one has a unique ID to be linked with the Java activity, but they also have unique IDs for each possible selection. The selected answers can then be acted upon by if statements using:

```
1 if (radioOne.getCheckedRadioButtonId() == R.id.radioY1)
2   { /* Do action */ }
```

5.9.11 Initialising AudioRecord

It is worth looking in further detail at the AudioRecord android class that we use for the audio input. The initialisation of new AudioRecord() uses five parameters, which are explained below:

- audioSource should be set to 1 to default to the inbuilt microphone
- sampleRateInHz should be set to 44100 to guarantee functionality on all devices, however a much lower sample rate would be preferable for the basic application here, and a solution to this is described below.
- channelConfig should be set to 16 for recording sound in MONO rather than STEREO.
- audioFormat should be set to 2 for PCM 16bit or 3 for PCM 8bit. This application uses 16bit to ensure functionality on the greatest number of devices possible.
- bufferSizeInBytes is set using an inbuilt method called getMinBufferSize(). The inputs to this are the same as parameters 2, 3 and 4 in the AudioRecord initialisation.

So the initialisation for the application should look as follows:

```
1 new AudioRecord(1,44100,16,2,getMinBufferSize(44100,16,2));
```

In practice, each input is assigned to a variable that is defined at the top of the code for easy modification and better clarity in the code.

For recording snoring frequencies, a sample rate of as low as 4000Hz would be adequate. The following

code solves the earlier discussed issue of minimising the sampling rate. The getMinBufferSize() is tested on each value added in the for() loop, but as soon as the function returns a positive (hence a valid) bufferSize, the method returns this lowest valid sample rate and breaks to finish the method.

```
1 public int getValidSampleRates() {
2   int samplerate = 0;
3 for (int rate : new int[] {8000, 11025, 16000, 22050, 44100}) {
4   int bufferSize = AudioRecord.getMinBufferSize(rate, ...
        AudioFormat.CHANNEL_CONFIGURATION_DEFAULT, AudioFormat.ENCODING_PCM_16BIT);
5 if (bufferSize > 0) {
6   samplerate = rate;
7   break;}
8 }
9 return samplerate };
```

5.9.12 Progress Wheel

A progress wheel requires two tasks to be carried out simultaneously the wheel itself, and the task that has the progress being indicated. This is done with the AsyncTask that allows a background operation to interact with the user interface element: effectively two threads are running and we can allow the UI thread to be updated from the background task. There are four self-explanatory steps in the asynchronous task called onPreExecute, doInBackground, onProgressUpdate and onPostExecute.

To use an example in this app, the SaveToFileTask first calls the progress wheel to show on screen with ProgressDialog.show() in the onPreExecute() step. It then calls the method audioData.saveData() in the doInBackground() step and finally calls progressDialog.dismiss() in onPostExecute() which will clearly indicate that the data has been saved. Note that the onProgressUpdate step is not used, as we do not have a use for updating the UI whilst running the background task. This might, for example, be used to change the text on the progress wheel to indicate when a second task is being carried out had one been present.

5.9.13 GraphView

GraphView is an additional library available for Android and is particularly useful for an audio application that looks at sound intensities.

5.9.14 Other Platforms

Apple provide their own libraries similar to those in android, that provide quickly usable methods and hardware capability to a developer. The sleep apnea app has suitably simple components that can be recreated on Apples iOS like for like if required. This section of the report will quickly overview the techniques to do this, but will also detail other available tools and how they might be used more to create a more effective native app in iOS. This section will not detail any code, purely discussion of the available libraries and tools.

5.9.15 Xcode

Xcode is the free software available in the Apple App Store that contains all the necessary prototyping and developing tools to produce an application on iPhone, iPad and Mac. Code can be written in Objective-C, C and a whole multitude of other languages with interpreters to adapt them. Xcode provides an Integrated Development Environment (IDE), compiler with inbuilt bug finding, an iOS simulator to virtually test the app amongst other things. Similarly to the Android Developer Tools, memory analysis, CPU profiling and general performance tracking is all included. This would provide a very suitable basis to build our app from.

5.9.16 AVAudioRecord Library

The most important library to consider here is the inbuilt capabilities of the audio recording. A handful of the more unique and interesting classes will be discussed here:

- recordAtTime:forDuration: This flexibility in setting the start time to record and the duration of each recording could be applied for a few things. Firstly the start time could be set such to greatly increase the chance of the user being asleep as the recording starts. It could also be used as a basic way to reduce background noise if, perhaps, the users sleeping area experienced background noise from parties or traffic until a set time in the evening. Likewise, limiting the duration of the recording could ensure results arent skewed from noise of waking up and manually turning the app off. The app could select a key range of hours in the night that maximises the probability of apnea detection and minimises the chance of background noise. Note that this could also be manually implemented in the Android version which is discussed later.
- averagePowerForChannel This is one of two values provided in the Audio Level Metering section

of the library. This could provide a very simple technique to implement a switching state model with perhaps three defined ranges of audio powers. Recording the average power for the channel (in our case there is only one channel in MONO recording) roughly twice per second would provide an array which would accurately reflect the volume of the sleeping user. Band two would represent the standard snoring volume observed in almost all sufferers of sleep apnea and would be the expected volume level for the user. Band one would be very low audio power indicative that the sleepers throat is blocked and an apnea is occurring. Band three would represent high audio power, interpreted as the loud snorting very commonly experienced after an apneic episode as the sleeper unblocks their throat. Clearly such a monitoring system would produce very small file sizes, which is a very desirable aim in audio recording based software (in this case, audio isnt being saved directly to a file, only monitored for power values). This technique doesnt lend itself well to machine learning techniques due to the lack of precision in the recorded data features arent defined nearly as well as more fully sampled techniques.

- peakPowerForChannel is the other value provided by the Audio Metering section and whilst not being as directly applicable, could be useful when used in conjunction with averagePowerForChannel. From simple logical reasoning, noise is much harder to detect when the precision of the system is low like the technique suggested above. peakPowerForChannel could therefore be used to calibrate this technique. A short recording of the background noise in controlled quiet conditions could calibrate the allowable background noise for band one in the above example, whilst the user recording a faked snort sound could provide a rough expectation for the power to be expected at band three. From this, unexpected audio power levels can be rejected or handled differently from the normal operational audio power.

Chapter 6

Conclusion

Obstructive Sleep Apneoa (OSA) is a serious and prevalent condition that affects many middle-aged people, and many of them remain undiagnosed, leading to poor sleep and a reduced quality of life. The current diagnosis solutions offered by the NHS are relatively expensive and inconvenient, and there is a need in the market for a way to check for OSA without having to spend a lot of time, money or effort. The easiest way to provide for this need would be through a smartphone application that would utilise machine learning to diagnose OSA from the user's audio sleep data.

After proving that there is a need and potential market for a mobile app that is able to provide a preliminary diagnosis of OSA, we moved on to first develop a method of diagnosis without using machine learning, to use as a backup and to familiarise ourselves with sleep data. We created a simple method of diagnosis in `MATLAB`® using power thresholds and variance analysis. This method was able to provide us with an output of 1s or 0s, highlighting different time periods when the user experiences apneoa. We then tested the simple method on a 7 minute audio file obtained from the internet, of a real OSA patient.

We then moved on to research several machine learning techniques that could be used in our app, and identified Support Vector Machines (SVM) and Hidden Markov Models (HMM) as being suitable in our case. We built the models in `MATLAB`® and ran tests using data obtained from PhysioNet to test the accuracy of both models. We presented the results and argued in Section 5.7.6 that the HMM model would be better suited for the purpose of diagnosing OSA from temporal data. More importantly, what we were able to show was that if the appropriate model is used, OSA can be accurately diagnosed using only a mobile application on a smartphone and there is the potential to develop and launch it as a complementary service to the current NHS diagnosis and treatment service for OSA. This would save even more time and resources for doctors as the NHS approved questionnaires can be easily performed

through the app even before meeting the GP.

In order to improve the accuracy of the diagnosis further, we need to improve both the quality and quantity of testing and training data. Firstly, using audio data (instead of the ECG data that we used in our initial experiments) would provide better applicability to the smartphone. Secondly, improving the quality, in terms of better annotations and multiple annotations by healthcare professionals to ensure accuracy of labelling would improve the accuracy of the model. Thirdly, simply providing more data for the model to estimate the parameters more accurately would, over time, make the app much better than it already is at identifying OSA in users. Lastly, the models we have developed can always be improved upon as well, as shown in Section 5.7.6.

We conclude with the assertion that the implementation and use of the mobile app we have developed would benefit the general public greatly, and also provide many benefits for healthcare professionals as well. We hope that further work will be done in this area to improve and refine the process, such that we finally find an app in the open market.

Appendix A

Code for the simple model

A.1 Main script – simple.m

```
% Reading data
clear all;
close all;
FILENAME = 'signal.wav';
[YRaw,f] = audioread(FILENAME) ;
TMAX = length(YRaw) / f;

% Sampling data because we don't need very high resolution
scale = 20;
for i = 1:(length(YRaw) / scale)
    Y(i) = 0.5 * (YRaw(i * scale, 1) + YRaw(i * scale, 2));
end

% Plots sampled data
figure;
plot(Y);
title('Sleep signal');

% Calculates and plots frequency
N = length(Y);
Yf = fft(Y,N);
freq = ((0:1/N:1-1/N)*f).'; % Frequency vector;
```

```
23
24 % Plot spectrum.
25 plot(freq,abs(Yf))
26 title('Amplitude Spectrum of y(t)')
27 xlabel('Frequency (Hz)')
28 ylabel('|Y(f)|')
29
30 % Calculates and plots the power
31 power = Y.^2;
32
33 % Apnea detection
34 sensitivityMean = 0.01;
35 interval = 3;
36 sensitivityVar = 5e-2;
37 windowSize = 20;
38 apnea = detectApnea(power, TMAX, sensitivityMean, interval);
39 apneaVar = detectApneaVar(power , TMAX , sensitivityVar , windowSize);
```

A.2 Detecting apnoea – detectApnea.m

```
1  function apnea = detectApnea(Y, TMAX, sensitivityMean, interval)
2      n = length(Y); % No. of samples
3      sampleInterval = round(n / TMAX * interval); % Minimum no. of consecutive samples ...
           that should be below threshold for apnea
4      apnea = zeros(1, n); % Generate a vector that will contain either zeros or ones ...
           to show where apnea is
5
6      threshold = mean(Y) / sensitivityMean;
7
8      nBelowThreshold = 0; % Number of sample points below the threshold in a row.
9      for i = 1:n
10         if Y(i) ≤ threshold
11             nBelowThreshold = nBelowThreshold + 1; % Increase # of sample points ...
                   below the threshold in a row
12         else % We see a point that is above threshold.
13             if nBelowThreshold ≥ sampleInterval % Condition for which we classify apnea.
```

```
14                  for j = max((i - nBelowThreshold), 1):(i - 1) % Loop through the last ...
                        'nBelowThreshold' points. Making sure (i - nBelowThreshold) ...
                        doesn't go below 1 (otherwise error).
15                      apnea(j) = 1;
16                  end
17              else
18                  for j = max((i - nBelowThreshold), 1):(i - 1)
19                      apnea(j) = 0;
20                  end
21              end
22              nBelowThreshold = 0;
23          end
24      end
25  end
```

A.3 Detecting apnoea using the 'variance' method – detectApneaVar.m

```
1  function apnea = detectApneaVar (Y , TMAX , sensitivityVar , windowSize)
2  n = length (Y);
3  apnea = zeros (1 , n);
4
5  % Calculate variance of whole signal (as a yardstick)
6  variance = var(Y) ;
7  maxVariance = variance / sensitivityVar;
8
9  % Calculate the variance through a moving window
10 twindow = [1:windowSize ; Y( 1:windowSize )]; % This vector contains the locations of ...
       the moving window
11
12 for i = 1:(n - windowSize)
13     sigma2 = var( twindow(2, :) );
14
15     if sigma2 > maxVariance
16         apnea(1 , twindow(1,:) ) = 1;
17     end
18
```

```
19      twindow(1,:) = twindow(1,:) + 1; % Update the window for the next round of ...
            variance calculations
20      twindow(2,:) = Y ( twindow(1,:) );
21  end
```

Appendix B

Code for reading data – readData.m

```
1  function [O, q, time, signal, annTime] = readData(fileIndex, keepSignal)
2  % Reads data from indices specified in the vector fileIndex and returns
3  %   O = observations, merged together [TxD]
4  %   q = latent states, 1-0 for apnea-nonapnea [Tx1]
5  % Inputs
6  %   fileIndex = vector of file indices to read
7  %   keepSignal = keep signal and time for plotting purposes (set to 1 or 0)
8
9      disp('Reading and conditioning data...');
10     filenames = getFilenames();
11
12     lastTime = 0;
13     annTimeLast = 0;
14     time = []; % time
15     signal = []; % signal
16     annTime = []; % timestamps of annotations
17     q = []; % latent states
18     O = []; % observations
19     for i = fileIndex
20         filename = cell2mat(filenames(i));
21         disp(sprintf('\tProcessing file %d: %s...', i, filename));
22
23         [timeTemp, signalTemp, freq] = rdsamp(filename); % reads the signal
24         [annTimeTemp, type] = rdann(filename,'apn'); % reads annotations
```

```
25          type = (type == 'A');
26
27          %% Conditioning data
28          annTimeTemp = [1; annTimeTemp];
29          q = [q; type]; % latent states
30          for i = 1:length(type)
31              O = [O; signalTemp((annTimeTemp(i) + 1):annTimeTemp(i + 1))'];
32          end
33
34          if keepSignal
35              nObs = annTimeTemp(end);
36              time = [time; timeTemp(1:nObs) + lastTime + 0.01];
37              annTime = [annTime; annTimeTemp(2:end) + annTimeLast];
38              signal = [signal; signalTemp(1:nObs)];
39              lastTime = time(end);
40              annTimeLast = annTime(end);
41          end
42      end
43      if ¬keepSignal
44          signal = [];
45          time = [];
46      end
47  end
```

Appendix C

Code for data conditioning

C.1 Transforming to spectrogram space – transformSpectrogram.m

```
function [OTransformed, qTransformed] = transformSpectrogram(O, q)
% O = matrix of T D-dimensional observations [TxD]
% q = vector of T hidden states [Tx1]
% OTransformed = spectrograms after transformation and cut-off [T'xD']
% qTransformed = hidden states after cut-off [T'x1]

    [T, D] = size(O);

    freq = 100;
    nBinsPerAnn = 6; % number of bins per annotation
    window = D / 2; % length of window
    nOverlap = window - D / nBinsPerAnn;
    OBig = reshape(O', 1, T * D);
    [Ss, Fs, Ts, Ps] = spectrogram(OBig, window, nOverlap, [], freq); % time on x ...
        axis, freq on y axis

    timeCutOff = (T - 1) * nBinsPerAnn;
    freqCutOff = 1500;
    Ps = Ps(1:freqCutOff, 1:timeCutOff);

    OTransformed = reshape(Ps, nBinsPerAnn * size(Ps, 1), T - 1)';
    qTransformed = q(1:(T - 1));
```

```
22 end
```

C.2 Calculating PCA parameters – pcaCalc.m

```
1  function [PCoef, PVec, xMean, xVar] = pcaCalc(x, k)
2  % Calculates the PCA parameters to reduce D-dimensional vectors stored in x
3  % [NxD] to K-dimensional vectors stored in y [NxK]
4  %   PCoef = vector of eigenvalues (variances)
5  %   PVec = k principal bases stored in k columns [DxK]
6  %   xMean = mean of data x (for future normalisation purposes)
7  %   xVar = var of data x (for future normalisation purposes)
8
9      % Normalise
10     xMean = mean(x);
11     xVar = var(x, 1);
12     x = bsxfun(@minus, x, xMean);
13     x = bsxfun(@rdivide, x, sqrt(xVar));
14
15     % PCA
16     [PCoef, PVec] = pca(x); % DxD, with i-th column being the i-th principal ...
           component (from PMTK3 package)
17     PVec = PVec(:, 1:k); % consider only the first k components
18 end
```

C.3 Reducing dimensions using PCA – pcaTransform.m

```
1  function y = pcaTransform(x, PVec, xMean, xVar)
2  % Reduces dimensions of D-dimensional vectors in x [NxD] to K-dimensional
3  % vectors stored in y [NxK] using PCA parameters
4
5      % Normalise
6      x = bsxfun(@minus, x, xMean);
7      x = bsxfun(@rdivide, x, sqrt(xVar));
```

```
 8
 9       % PCA
10       y = x * PVec;
11   end
```

Appendix D

Code for Support Vector Machines

D.1 Main script – apneaSVM.m

```
trainIndex = 1:10;
testIndex = 11:15;

%% Reading and conditioning data
[X, Y, time, signal, annTime] = readData(trainIndex, 1); % X = input, Y = output

%% Transform spectrogram
[XTransformed, YTransformed] = transformSpectrogram(X, Y);

%% PCA
disp('PCA...');
k = 100; % take k principal components only
[PCoef, PVec, XMean, XVar] = pcaCalc(XTransformed, k);
XPca = pcaTransform(XTransformed, PVec, XMean, XVar);

%% SVM
options = optimset('MaxIter', 1e6);
SVMStructPoly = svmtrain(XPca, YTransformed, 'kernel_function', 'polynomial', ...
    'polyorder', 3, 'options', options); % poly kernel
% SVMStructRbf = svmtrain(XPca, YTransformed, 'kernel_function', 'rbf', 'options', ...
    options); % rbf kernel
% SVMStruct = svmtrain(XPca, YTransformed, 'options', options); % no kernel
```

```
21
22 %% Comparing with test data
23 apneaSVMTest(SVMStruct, PVec, XMean, XVar, testIndex);
```

D.2 Test script – apneaSVMTest.m

```
1  function apneaSVMTest(SVMStruct, PVec, XMean, XVar, testIndex)
2      disp('Comparing with test data...');
3      accuracySum = 0;
4      for i = testIndex
5          %% Read data
6          [XTest, YTest, timeTest, signalTest, annTimeTest] = readData(i, 1);
7
8          %% Transform spectrogram
9          [XTest, YTest] = transformSpectrogram(XTest, YTest);
10
11         %% PCA
12         XTest = pcaTransform(XTest, PVec, XMean, XVar);
13
14         %% SVM Classify
15         YPredict = svmclassify(SVMStruct, XTest);
16
17         accuracy = sum(YPredict == YTest) / length(YTest)
18         accuracySum = accuracySum + accuracy;
19
20         %% Plotting data
21         ...
22     end
23     avgAccuracy = accuracySum / length(testIndex)
24 end
```

Appendix E

Code for Hidden Markov Models

E.1 Main script – apneaHMM.m

```
trainIndex = 1:10;
testIndex = 11:15;

%% Reading and conditioning data
[Y, X] = readData(trainIndex, 0); % Y = observations, X = latent states.
N = 2; % number of states: apnea-noapnea
S = [0; 1]; % set of possible states

%% Transform spectrogram
[YTransformed, XTransformed] = transformSpectrogram(Y, X);

%% PCA
disp('PCA...');
k = 100; % take k principal components only
[PCoef, PVec, YMean, YVar] = pcaCalc(YTransformed, k);
figure;
plot(cumsum(PCoef)/sum(PCoef));
YPca = pcaTransform(YTransformed, PVec, YMean, YVar);

%% Fitting parameters
[A, Mu, U, pi] = apneaHMMTrain(XTransformed, YPca, N, S);

```

```
23 %% Comparing with test data
24 apneaHMMTest(A, Mu, U, pi, PVec, YMean, YVar, N, testIndex);
```

E.2 Training script – apneaHMMTrain.m

```
1  function [A, Mu, U, pi] = apneaHMMTrain(q, O, N, S)
2      [T, D] = size(O); % dimension of observations & number observations
3
4      % Transitional matrix A
5      disp('Fitting transitions...');
6      A = zeros(N);
7      for i = 1:N
8          for j = 1:N
9              transitionsFromSiToSj = 0;
10             transitionsFromSi = 0;
11             for t = 1:(T - 1)
12                 transitionsFromSiToSj = transitionsFromSiToSj + (q(t) == S(i) && q(t ...
                       + 1) == S(j));
13                 transitionsFromSi = transitionsFromSi + (q(t) == S(i));
14             end
15             A(i, j) = transitionsFromSiToSj / transitionsFromSi;
16         end
17     end
18
19     % Estimate Gaussian parameters
20     disp('Fitting emissions...');
21     Mu = zeros(D, N); % means
22     U = zeros(D, D, N); % covariances
23     for j = 1:N
24         x = O(q == S(j), :);
25         model = gaussFit(x); % from PMTK3 package
26         Mu(:, j) = model.mu;
27         U(:, :, j) = model.Sigma;
28     end
29
30     % Estimate HMM parameters
```

```
31      pi = [0.5; 0.5];
32  end
```

E.3 Test script – apneaHMMTest.m

```
1   function apneaHMMTest(A, Mu, U, pi, PVec, YMean, YVar, N, testIndex)
2       disp('Comparing with test data...');
3       accuracySum = 0;
4       for i = testIndex
5           %% Read data
6           [YTest, XTest, timeTest, signalTest, annTimeTest] = readData(i, 1);
7
8           %% Transform spectrogram
9           [YTest, XTest] = transformSpectrogram(YTest, XTest);
10
11          %% PCA
12          YTest = pcaTransform(YTest, PVec, YMean, YVar);
13
14          %% Most likely path (Viterbi)
15          disp(sprintf('\tCalculating the most likely path...'));
16          B = [];
17          for j = 1:N
18              B(j, :) = gaussian_prob(YTest', Mu(:, j), U(:, :, j)); % from HMM Toolbox
19          end
20          path = viterbi_path(pi, A, B) - 1; % from PMTK3 package
21
22          accuracy = sum(path == XTest') / length(XTest)
23          accuracySum = accuracySum + accuracy;
24
25          %% Plotting data
26          ...
27      end
28      avgAccuracy = accuracySum / length(testIndex)
29  end
```

Appendix F

Division of labour

The work has been split in the following way.

- George Cochrane has written 4 The app, and 5.9 Designing an Android App.
- Tuan Anh Le has written 5.6 Machine Learning – Theory, 5.7.2 Conditioning input data, 5.7.3 Data Visualisation, 5.7.4 Support Vector Machines, 5.8 Machine Learning – Java, C Code for data conditioning and D Code for Support Vector Machines.
- Sophie Louth has written 2 The Condition, 3.1 Medical Rational, 5.1 Diagnostic Approaches, 5.2 Symptoms of OSA, 5.3 Questionnaire selection, and 5.4 Audio
- Sachin Mylavarapu has written 3.2 Economic Rationale, 5.5 Signal Analysis – simple method, 5.7.5 Hidden Markov Models, 5.7.6 Summary, A Code for the simple model, B Code for reading data – readData.m, and E Code for Hidden Markov Models.

Bibliography

[1] Nancy Adams, Milton Strauss, Mark Schluchter, and Susan Redline. Relation of measures of sleep-disordered breathing to neuropsychological functioning. *American journal of respiratory and critical care medicine*, 163(7):1626–1631, 2001.

[2] Sonia Ancoli-Israel, Daniel F Kripke, Melville R Klauber, William J Mason, Robert Fell, and Oscar Kaplan. Sleep-disordered breathing in community-dwelling elderly. *Sleep*, 14(6):486, 1991.

[3] Apneos. Obstructive sleep apnea : Details, other factors, 2003.

[4] American Sleep Disorders Association et al. International classification of sleep disorders: Diagnostic and coding manual. revised ed, 2001.

[5] Ali Azarbarzin and Zahra Moussavi. Unsupervised classification of respiratory sound signal into snore/no-snore classes. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 3666–3669. IEEE, 2010.

[6] Carol M Baldwin, Kent A Griffith, F Javier Nieto, George T O'Connor, Joyce A Walsleben, and Susan Redline. The association of sleep-disordered breathing and sleep symptoms with quality of life in the sleep heart health study. *Sleep*, 24(1):96–105, 2001.

[7] Ethan M Balk, Mei Chung, Denish Moorthy, Jeffrey A Chan, Kamal Patel, Thomas W Concannon, Sara J Ratichek, and Lina Kong Win Chang. Future research needs for diagnosis of obstructive sleep apnea. 2012.

[8] EUGENI BALLESTER, JOAN R BADIA, LOURDES HERNANDEZ, EVA CARRASCO, JUAN de PABLO, CONSOL FORNAS, ROBERT RODRIGUEZ-ROISIN, and JOSEP M MONTSERRAT. Evidence of the effectiveness of continuous positive airway pressure in the treatment of sleep

apnea/hypopnea syndrome. *American journal of respiratory and critical care medicine*, 159(2):495–501, 1999.

[9] Nir Ben-Israel, Ariel Tarasiuk, Yaniv Zigel, et al. Obstructive apnea hypopnea index estimation by analysis of nocturnal snoring signals in adults. *Sleep*, 35(9):1299–305C, 2012.

[10] Camil Castelo-Branco, Santiago Palacios, Javier Ferrer-Barriendos, and Xavier Alberich. Do patients lie? an open interview vs. a blind questionnaire on sexuality. *The journal of sexual medicine*, 7(2pt2):873–880, 2010.

[11] M Cavusoglu, M Kamasak, O Erogul, T Ciloglu, Y Serinagaoglu, and T Akcam. An efficient method for snore/nonsnore classification of sleep sounds. *Physiological measurement*, 28(8):841, 2007.

[12] NHS Choices. Diagnosing sleep apnoea, 2012.

[13] NHS Choices. Obstructive sleep apnoea/hypopnoea - suspected, 2014.

[14] F Chung, R Subramanyam, P Liao, E Sasaki, C Shapiro, and Y Sun. High stop-bang score indicates a high probability of obstructive sleep apnoea. *British journal of anaesthesia*, 108(5):768–775, 2012.

[15] Frances Chung, Balaji Yegneswaran, Pu Liao, Sharon A Chung, Santhira Vairavanathan, Sazzadul Islam, Ali Khajehdehi, and Colin M Shapiro. Stop questionnaire: a tool to screen patients for obstructive sleep apnea. *Anesthesiology*, 108(5):812–821, 2008.

[16] Frances Chung, Balaji Yegneswaran, Pu Liao, Sharon A Chung, Santhira Vairavanathan, Sazzadul Islam, Ali Khajehdehi, and Colin M Shapiro. Validation of the berlin questionnaire and american society of anesthesiologists checklist as screening tools for obstructive sleep apnea in surgical patients. *Anesthesiology*, 108(5):822–830, 2008.

[17] David F Dinges. A survey screen for prediction of apnea. *Sleep*, 18(3):158–166, 1995.

[18] John B Dixon, Linda M Schachter, et al. Predicting sleep apnea and excessive day sleepiness in the severely obeseindicators for polysomnography. *CHEST Journal*, 123(4):1134–1141, 2003.

[19] eMarketer.com. Smartphone users worldwide will total 1.75 billion, January 2014.

[20] Public Health England. Adult obesity, 2014.

[21] Heather M Engleman, Ruth N Kingshott, Peter K Wraith, Thomas W Mackay, Ian J Deary, and Neil J Douglas. Randomized placebo-controlled crossover trial of continuous positive airway pressure for mild sleep apnea/hypopnea syndrome. *American Journal of Respiratory and Critical Care Medicine*, 159(2):461–467, 1999.

[22] Paul L Enright, Anne B Newman, Patricia W Wahl, Teri A Manolio, Edward F Haponik, and PJ Boyle. Prevalence and correlates of snoring and observed apneas in 5,201 older adults. *Sleep*, 19(7):531–538, 1996.

[23] Lawrence J Epstein, David Kristo, Patrick J Strollo Jr, Norman Friedman, Atul Malhotra, Susheel P Patil, Kannan Ramar, Robert Rogers, Richard J Schwab, Edward M Weaver, et al. Clinical guideline for the evaluation, management and long-term care of obstructive sleep apnea in adults. *J Clin Sleep Med*, 5(3):263–276, 2009.

[24] Larry J Findley, Michael J Fabrizio, Herbert Knight, Barbara B Norcross, Amy J Laforte, and Paul M Suratt. Driving simulator performance in patients with sleep apnea. *American Review of Respiratory Disease*, 140(2):529–530, 1989.

[25] Larry J Findley, Mark E Unverzagt, and Paul M Suratt. Automobile accidents involving patients with obstructive sleep apnea. *American Review of Respiratory Disease*, 138(2):337–340, 1988.

[26] Kevin J Finkel, Adam C Searleman, Heidi Tymkew, Christopher Y Tanaka, Leif Saager, Elika Safer-Zadeh, Michael Bottros, Jacqueline A Selvidge, Eric Jacobsohn, Debra Pulley, et al. Prevalence of undiagnosed obstructive sleep apnea among adult surgical patients in an academic medical center. *Sleep medicine*, 10(7):753–758, 2009.

[27] Laurel Finn, Terry Young, Mari Palta, and Dennis G Fryback. Sleep-disordered breathing and self-reported general health status in the wisconsin sleep cohort study. *SLEEP-NEW YORK-*, 21:701–708, 1998.

[28] JA Fiz, J Abad, R Jane, M Riera, MA Mananas, P Caminal, D Rodenstein, and J Morera. Acoustic analysis of snoring sound in patients with simple snoring and obstructive sleep apnoea. *European Respiratory Journal*, 9(11):2365–2370, 1996.

[29] W WARD Flemons, William A Whitelaw, Rollin Brant, and John E Remmers. Likelihood ratios for a sleep apnea clinical prediction rule. *American journal of respiratory and critical care medicine*, 150(5):1279–1285, 1994.

[30] RB Fogel, A Malhotra, and DP White. Sleep· 2: Pathophysiology of obstructive sleep apnoea/hypopnoea syndrome. *Thorax*, 59(2):159–163, 2004.

[31] British Lung Foundation. The osa charter, 2014.

[32] Karl A Franklin, Per Ake Holmgren, Fredrik Jonsson, Nils Poromaa, Hans Stenlund, and Eva Svanborg. Snoring, pregnancy-induced hypertension, and growth retardation of the fetus. *CHEST Journal*, 117(1):137–141, 2000.

[33] Jean-Marc Franois. jahmm: An implementation of hidden markov models in java. `https://code.google.com/p/jahmm/`, 2006.

[34] Shawn D Gale and Ramona O Hopkins. Effects of hypoxia on the brain: neuroimaging and neuropsychological findings following carbon monoxide poisoning and obstructive sleep apnea. *Journal of the International neuropsychological society*, 10(1):60–71, 2004.

[35] Elnaz Gederi and Gari D Clifford. Fusion of image and signal processing for the detection of obstructive sleep apnea. In *Biomedical and Health Informatics (BHI), 2012 IEEE-EMBS International Conference on*, pages 890–893. IEEE, 2012.

[36] Zoubin Ghahramani and Geoffrey E Hinton. Variational learning for switching state-space models. *Neural computation*, 12(4):831–864, 2000.

[37] Hirotaka Hara, Naoko Murakami, Yuji Miyauchi, and Hiroshi Yamashita. Acoustic analysis of snoring sounds by a multidimensional voice program. *The Laryngoscope*, 116(3):379–381, 2006.

[38] A.M. Howatson, P.G. Lund, J.D. Todd, P.D. McFadden, P.J.P. Smith, and University of Oxford. Department of Engineering Science. *Engineering Tables and Data*. University of Oxford, Department of Engineering Science, 2008.

[39] Conrad Iber, American Academy of Sleep Medicine, et al. *The AASM manual for the scoring of sleep and associated events: rules, terminology and technical specifications*. American Academy of Sleep Medicine, 2007.

[40] Mary SM Ip, Bing Lam, Ian J Lauder, Kenneth WT Tsang, Ka-fai Chung, Yuk-wan Mok, and Wah-kit Lam. A community study of sleep-disordered breathing in middle-aged chinese men in hong kong. *CHEST Journal*, 119(1):62–69, 2001.

[41] Wall Street Journal. A test for sleep apnea from your own bedroom, May 2013.

[42] Hyon C Kim, Terry Young, Charles G Matthews, Steven M Weber, Austin R Woodard, and Mari Palta. Sleep-disordered breathing and neuropsychological deficits: a population-based study. *American journal of respiratory and critical care medicine*, 156(6):1813–1819, 1997.

[43] Simon D Kirby, P Eng, Wayne Danter, Charles FP George, Tanya Francovic, Ralph RF Ruby, and Kathleen A Ferguson. Neural network prediction of obstructive sleep apnea from clinical criteria. *CHEST Journal*, 116(2):409–415, 1999.

[44] Vladimir Kostyukov. la4j: Linear algebra for java. `http://la4j.org/`, 2011-2013.

[45] Clete A Kushida, Bradley Efron, and Christian Guilleminault. A predictive morphometric model for the obstructive sleep apnea syndrome. *Annals of Internal Medicine*, 127(8_Part_1):581–587, 1997.

[46] leahthegeek. Nick sleep apnea proof 1, 2007. Online Video - Accessed 8th January 2014.

[47] JC Leiter. Upper airway shape: Is it important in the pathogenesis of obstructive sleep apnea? *American journal of respiratory and critical care medicine*, 153(3):894–898, 1996.

[48] Atul Malhotra and David P White. Obstructive sleep apnoea. *The lancet*, 360(9328):237–245, 2002.

[49] R Martin. Labelled sagittal view of face, 2014.

[50] MedIndia. Manipal university and xerox innovation to test remote-sensing healthcare technologies, February 2014.

[51] Mark B Mengel, Warren Lee Holleman, and Scott A Fields. *Fundamentals of clinical practice*. Springer, 2002.

[52] Doris Moser, Peter Anderer, Georg Gruber, Silvia Parapatics, Erna Loretz, Marion Boeck, Gerhard Kloesch, Esther Heller, Andrea Schmidt, Heidi Danker-Hopfe, et al. Sleep classification according to aasm and rechtschaffen & kales: effects on sleep scoring parameters. *Sleep*, 32(2):139, 2009.

[53] Kevin P Murphy. *Machine learning: a probabilistic perspective*. Cambridge, MA, 2012.

[54] Lung National Heart and Blood Institute. Other names for sleep apnea, 2012.

[55] Nikolaus C Netzer, Riccardo A Stoohs, Cordula M Netzer, Kathryn Clark, and Kingman P Strohl. Using the berlin questionnaire to identify patients at risk for the sleep apnea syndrome. *Annals of internal medicine*, 131(7):485–491, 1999.

[56] Andrew Ng. Cs229 - lecture notes. Course website, 2013.

[57] Andrew Keong Ng, Tong San Koh, Eugene Baey, Teck Hock Lee, Udantha Ranjith Abeyratne, and Kathiravelu Puvanendran. Could formant frequencies of snore signals be an alternative means for the diagnosis of obstructive sleep apnea? *Sleep medicine*, 9(8):894–898, 2008.

[58] Andrew Keong Ng, Tong San Koh, Udantha Ranjith Abeyratne, and Kathiravelu Puvanendran. Investigation of obstructive sleep apnea using nonlinear mode interactions in nonstationary snore signals. *Annals of biomedical engineering*, 37(9):1796–1806, 2009.

[59] Andrew Keong Ng, Tong San Koh, Kathiravelu Puvanendran, and Udantha R Abeyratne. Snore signal enhancement and activity detection via translation-invariant wavelet transform. *Biomedical Engineering, IEEE Transactions on*, 55(10):2332–2342, 2008.

[60] Andrew Keong Ng, Kim Yong Wong, Chin Hao Tan, and Tong San Koh. Bispectral analysis of snore signals for obstructive sleep apnea detection. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 6195–6198. IEEE, 2007.

[61] NHS. Sleep apnoea - diagnosis, June 2012.

[62] NHS. Sleep apnoea, June 2014.

[63] Novasom. Economics of home sleep testing, 2014.

[64] Novasom. Why accusom?, 2014.

[65] NPR. Apps for apnea? new gadgets promise to improve sleep, February 2012.

[66] University of Georgia. Basic strategies for diagnosis, 2014.

[67] Department of Health. Confidentiality: Nhs code of practice, 2003.

[68] Wulf Pankow, Achim Lies, Friedrich W Lohmann, et al. Sleep-disordered breathing and hypertension. *N Engl J Med*, 343(13):966, 2000.

[69] David Knott Steve Van Kuiken Peter Groves, Basel Kayyali. The 'big data' revolution in health care. *McKinsey & Company*, 2013.

[70] PhysioNet. Detecting and quantifying apnea based on the ecg, 2012.

[71] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[72] Ewaryst Rafajłowicz, Miroslaw Pawlak, and Ansgar Steland. Nonparametric sequential change-point detection by a vertically trimmed box method. *Information Theory, IEEE Transactions on*, 56(7):3621–3634, 2010.

[73] Daniel Ramage. Cs229 - section notes. Course website, 2007.

[74] Manu Rastogi, Dipankar Nagchoudhuri, and CD Parikh. Quadratic phase coupling in ecg signals. In *Sensors and the International Conference on new Techniques in Pharmaceutical and Biomedical Research, 2005 Asian Conference on*, pages 74–77. IEEE, 2005.

[75] Allan Rechtschaffen and Anthony Kales. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects. 1968.

[76] S Roberts and L Tarassenko. New method of automated sleep quantification. *Medical and Biological Engineering and Computing*, 30(5):509–517, 1992.

[77] A Roebuck, V Monasterio, E Gederi, M Osipov, J Behar, A Malhotra, T Penzel, and GD Clifford. A review of signals used in sleep analysis. *Physiological measurement*, 35(1):R1, 2014.

[78] Wendy A Rogers. Is there a moral duty for doctors to trust patients? *Journal of medical ethics*, 28(2):77–80, 2002.

[79] Leon D Rosenthal and Diana C Dolan. The epworth sleepiness scale in the identification of obstructive sleep apnea. *The Journal of nervous and mental disease*, 196(5):429–431, 2008.

[80] Alex Brandao Rossow, Evandro Ottoni Teatini Salles, and Klaus Fabian Coco. Automatic sleep staging using a single-channel eeg modeling by kalman filter and hmm. In *Biosignals and Biorobotics Conference (BRC), 2011 ISSNIP*, pages 1–6. IEEE, 2011.

[81] Staples High School. Labelled anterior view of mouth, 2014.

[82] Richard J Schwab, Krishanu B Gupta, Warren B Gefter, Louis J Metzger, Eric A Hoffman, and Allan I Pack. Upper airway and soft tissue anatomy in normal subjects and patients with sleep-disordered breathing. significance of the lateral pharyngeal walls. *American Journal of Respiratory and Critical Care Medicine*, 152(5):1673–1689, 1995.

[83] Joan LF Shaver and Shannon N Zenk. Review: Sleep disturbance in menopause. *Journal of women's health & gender-based medicine*, 9(2):109–118, 2000.

[84] J Sola-Soler, R Jane, JA Fiz, and J Morera. Spectral envelope analysis in snoring signals from simple snorers and patients with obstructive sleep apnea. In *Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE*, volume 3, pages 2527–2530. IEEE, 2003.

[85] Jordi Solà-Soler, José Antonio Fiz, José Morera, and Raimon Jané. Multiclass classification of subjects with sleep apnoea–hypopnoea syndrome through snoring analysis. *Medical engineering & physics*, 34(9):1213–1220, 2012.

[86] JR Stradling, JH Crosby, and CD Payne. Self reported snoring and daytime sleepiness in men aged 35-65 years. *Thorax*, 46(11):807–810, 1991.

[87] Kingman P Strohl and Susan Redline. Recognition of obstructive sleep apnea. *American journal of respiratory and critical care medicine*, 154(2):279–289, 1996.

[88] The Economic Times. Xerox, manipal to develop non-contact diagnostics solutions, February 2014.

[89] Willis H Tsai, John E Remmers, Rollin Brant, W Ward Flemons, Jan Davies, and Colin Macarthur. A decision rule for diagnostic testing in obstructive sleep apnea. *American journal of respiratory and critical care medicine*, 167(10):1427–1432, 2003.

[90] DL Van Brunt, Kenneth L Lichstein, Sharon L Noe, RN Aguillard, and Kristin W Lester. Intensity pattern of snoring sounds as a predictor for sleep-disordered breathing. *Sleep*, 20(12):1151–1156, 1997.

[91] B Ph Van Milligen, E Sanchez, T Estrada, C Hidalgo, B Branas, B Carreras, and L Garcia. Wavelet bicoherence: a new turbulence analysis tool. *Physics of Plasmas (1994-present)*, 2(8):3017–3032, 1995.

[92] Sidney Viner, John P Szalai, and Victor Hoffstein. Are history and physical examination a good screening test for sleep apnea? *Annals of internal medicine*, 115(5):356–359, 1991.

[93] Ingrid Waldron. What do we know about causes of sex differences in mortality? a review of the literature. *Population Bulletin of the United Nations*, (18):59, 1985.

[94] WA Whitelaw. Characteristics of the snoring noise in patients with and without occlusive sleep apnea. *Am Rev Respir Dis*, 147:635–644, 1993.

[95] Wikipedia. Discrete fourier transform — Wikipedia, the free encyclopedia, 2014. [Online; accessed 28-March-2014].

[96] Wikipedia. Expectation-maximization algorithm — Wikipedia, the free encyclopedia, 2014. [Online; accessed 25-March-2014].

[97] Wikipedia. Hidden markov model — Wikipedia, the free encyclopedia, 2014. [Online; accessed 24-March-2014].

[98] Wikipedia. Machine learning — Wikipedia, the free encyclopedia, 2014. [Online; accessed 18-March-2014].

[99] Wikipedia. Spectral density — Wikipedia, the free encyclopedia, 2014. [Online; accessed 14-April-2014].

[100] Kent Wilson, Riccardo A Stoohs, Thomas F Mulrooney, Linda J Johnson, Christian Guilleminault, and Zhen Huang. The snoring spectrumacoustic assessment of snoring sound intensity in 1,139 individuals undergoing polysomnography. *CHEST Journal*, 115(3):762–770, 1999.

[101] Jacquet Wong. musicg: Lightweight java api for audio analysing. `https://code.google.com/p/musicg/`, 2014.

[102] Azadeh Yadollahi and Zahra Moussavi. Formant analysis of breath and snore sounds. In *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pages 2563–2566. IEEE, 2009.

[103] Terry Young. Sleep-disordered breathing in older adults: is it a condition distinct from that in middle-aged adults? *Sleep*, 19(7):529, 1996.

[104] Terry Young, Paul E Peppard, and Daniel J Gottlieb. Epidemiology of obstructive sleep apnea: a population health perspective. *American journal of respiratory and critical care medicine*, 165(9):1217–1239, 2002.

[105] Terry Young, Eyal Shahar, F Javier Nieto, Susan Redline, Anne B Newman, Daniel J Gottlieb, Joyce A Walsleben, Laurel Finn, Paul Enright, and Jonathan M Samet. Predictors of sleep-disordered breathing in community-dwelling adults: the sleep heart health study. *Archives of Internal Medicine*, 162(8):893–900, 2002.