An Efficient Algorithm to Measure Quality and Quantity of Sleep using Smartphone

Student Name: Alvika Gautam

IIIT-D-MTech-CS-MUC-12-061

May 10, 2014

Indraprastha Institute of Information Technology
New Delhi

Thesis Committee
Dr. Vinayak S. Naik (Advisor)
Dr. P.B. Sujit (Internal Examiner)
Mr. Dipyaman Banerjee (External Examiner)

Submitted in partial fulfillment of the requirements
for the Degree of M.Tech. in Computer Science,
with specialization in Mobile and Ubiquitous Computing

Keywords: Mobile, Sensing, Sleep Disorder, Sleep Apnea and HMM

Certificate

This is to certify that the thesis titled **'An Efficient Algorithm to Measure Quality and Quantity of Sleep using Smartphone'** submitted by **Alvika Gautam** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science & Engineering* is a record of the bonafide work carried out by her under my guidance and supervision in the Mobile and Ubiquitous computing group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Vinayak S. Naik
Indraprastha Institute of Information Technology, New Delhi

Abstract

Sleep quality and quantity affects an individual's personal health. Polysomnography (PSG) is the conventional approach for sleep monitoring. PSG has several limitations in terms of sleep monitoring by a regular user. A number of wearable devices with embedded sensors have emerged in recent past as an alternative to polysomnography (PSG) for regular sleep monitoring directly by the user. These devices are intrusive and cause user discomfort besides being expensive. We present an algorithm to detect sleep using a smartphone with the help of its inbuilt accelerometer sensor.

We exploit three different approaches to classify data into two states, Sleep and Wake. The approaches vary in terms of their adaptability. One is based on Kushida equation and uses a fixed threshold. Second is based on statistical method and also uses a fixed threshold but the threshold is decided dynamically at the time of data processing. Finally the third is based on Hidden Markov Model (HMM) training. The first one being least adaptable, the second one is moderate, and the third one is the most adaptive. The complexity also increases in the same order. Dataset consists of sleep data for four subjects for twelve days each collected using our android application for data collection. The accuracy of sleep detection of each of the three approaches is then compared with that of Zeo sensor, which is a medically approved device to measure sleep. We find that HMM training classification approach is as much as 84% accurate. It is more accurate as compared to Kushida equation based on fixed threshold and statistical method based on fixed threshold. In order to concisely represent the sleep quality of a person, we model the sleep data of a user using HMM. The parameters obtained from HMM can further be used to concisely represent sleep quality of an individual in the form of a model specific to that individual. We present an analysis to find out tradeoff between the amount of training data and the accuracy provided in modeling the sleep. We find that six days of sleep data is sufficient for accurate modeling. We compare accuracy of our HMM training algorithm with a representative third party application SleepTime available from Google Play Store for Android. We find that the classification done using HMM approach is closer to that done by Zeo by 13% as compared to the third party Android application SleepTime. In order to get an estimate as to how our classification algorithm will perform on a smartphone we compare the performance of our algorithm on two different PC architectures one with more and the other with less resources. Results suggest that the algorithm is feasible enough for execution on an android smartphone.

Acknowledgments

I would like to thank Dr. Vinayak S. Naik for his invaluable guidance and continuous motivation for doing this research work. I also thank IIIT-Delhi for providing me the full support for doing the work. I thank my fellow researcher Shreyasi Aggarwal and all other students for their help and insightful thoughts for doing the work. I thank all faculty members of IIIT-Delhi and my M.Tech colleagues for their encouragement and motivation in the past two years. Above all, I express my gratitude to my parents, family members and close friends who are a constant source of inspiration for me.

Alvika Gautam

Contents

1 Introduction **1**

1.1 Background 1

1.2 Aim of the Thesis 2

1.3 Organization of the thesis 3

2 Related Work **4**

3 Solution Approach **7**

3.1 Modelling of sleep pattern: General Concepts 7

3.1.1 Hidden Markov Model 8

3.1.2 Parameter Learning: Expectation Maximization for HMMs 9

3.1.3 HMM Viterbi Algorithm 9

3.2 Experimental Setup and Data collection 10

3.2.1 Experimental Setup 10

3.2.2 Our Android Data Collection Application 11

3.2.3 Data Collection 11

3.3 Techniques to Classify Sleep/Wake States 13

3.3.1 Data extraction from zeo for comparison 13

3.3.2 Classification using Kushida Equation based on Fixed Threshold 14

3.3.3 Classification using Statistical Method based on Fixed Threshold 14

3.3.4 Classification using HMM Training 15

4 Analysis and Results **16**

4.1 Comparison of Accuracy in Classifying Sleep/Wake States 16

4.2 Analysis of Technique for Modeling of Sleep Pattern 18

4.3 Comparison of obtained HMM parameters 20

4.4 Comparison With Third Party SleepTime Android Application 21

4.5 Performance of Our Algorithm 21

4.6 Performance of our Android Application 22

5 Conclusion **23**

6 Future work **24**

List of Figures

1.1 Placement of electrodes to determine EEG, EOG, and EMG. 2

3.1 HMM state transition diagram of Sleep-Wake process 8

3.2 Acceleration axes relative to the phone 10

3.3 Experimental setup. This shows the data collection on two Android phones along with the Zeo sensor. One phone receives the data collected by Zeo using Bluetooth and the other phone had the third party app and our app running. 11

3.4 Our Android application to collect data 12

3.5 Data collection showing raw sleep activity data obtained from accelerometer . . . 13

3.6 Output graph from Zeo sensor classifying sleep into REM, light, deep, wake along with number of times woken and total sleep. 13

4.1 Comparison of classified data plots using Zeo, Kushida equation, Statistical method, and HMM training approach for three days. The readings are consistent across the three representative days. 18

4.2 Evaluation of HMM approach showing the classification and posterior probability plots for 2, 6, and 11 days of training data respectively. Classification accuracy was 69%, 84% and 85% for 2, 6 and 11 days of training data respectively. 19

4.3 Output of the third party Android application : SleepTime 19

4.4 Comparison of emission probability obtained from HMM modeling for two subjects of different age groups (a) 21-25 years (b) 40-44 years 20

4.5 Comparison of power consumption against different accelerometer sampling rates along with baseline consumption of the phone 22

List of Tables

4.1 Comparison of approaches that classify Sleep/Wake states 17
4.2 Performance of our algorithm on two computer architectures 21

Chapter 1

Introduction

1.1 Background

Sleep is a behavioral state that is a natural part of every individual's life. We spend about one-third of our lives asleep.Quality and quantity of sleep play an important role in various aspects of physical, mental, and emotional health of a person. Poor sleeping habits/patterns may result in cardiovascular diseases and poor mental health, such as depression, stress, anxiety, etc. In current scenario and lifestyle sleep disorders such as sleep apnea, insomnia, hypersomnia [1] is getting increasingly common in all age groups which enhances the importance of study of sleep process.

One of the common choices to diagnose sleep disorders is PSG studies. These studies involve comprehensive recording of the bio-physiological changes that occur during sleep. The PSG monitors many body functions including limb movement, brain (EEG), eye movements (EOG), muscle activity, skeletal muscle activation (EMG), and heart rhythm (ECG) during sleep [20]. Previous research [16] has established limb movements and EEG as vital parameters in measuring sleep quantity and quality. To record these movements electrodes are placed on the scalp in a symmetrical pattern. The electrodes measure very small voltages that are caused by synchronized activity in very large numbers of synapses (nerve connections) in the brains outer layers (cerebral cortex). EEG data are represented by curves that are classified according to their frequencies. The wavy lines of the EEG are called brain waves. An electrooculogram (EOG) uses electrodes on the skin near the eye to measure changes in voltage as the eye rotates in its socket. The electrical activity associated with active muscles by using electromyograms (EMGs). In this technique, electrodes are placed on the skin overlaying a muscle. In humans, the electrodes are placed under the chin because muscles in this area demonstrate very dramatic changes during the various stages of sleep.
In practice, EEGs, EOGs, and EMGs are recorded simultaneously on continuously moving chart paper or digitized by a computer and displayed on a high-resolution monitor. This allows the relationships among the three measurements to be seen immediately. The patterns of activity in these three systems provide the basis for classifying the different types of sleep.

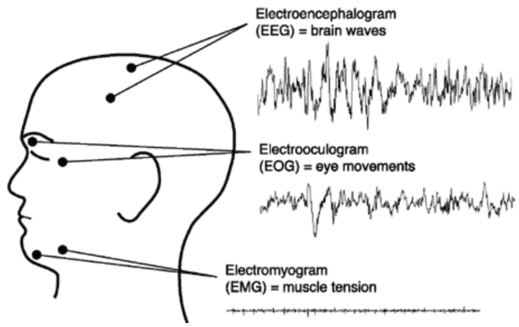


Figure 1.1: Placement of electrodes to determine EEG, EOG, and EMG.

In the 1970s a sleep disorder called sleep apnea [2] was identified and the breathing functions respiratory airflow and respiratory effort indicators were added along with peripheral pulse oximetry to PSG.

Collection of sleep disorder related data using the PSG tests requires a patient to be admitted in a hospital. These tests severely limit regular long term monitoring owing to the cost involved and the complexity arising due to hospital admission. Further, the data collected in these tests may not be representative of the patient's sleep pattern in her home under regular conditions. Advances permit sensors to be embedded within wristbands, bracelets, and belts and to wirelessly send data to a mobile computing device that can use the signals to make inferences. A number of commercial wearable devices have emerged in recent times, as an alternative to polysomnography (PSG) which enable the user to monitor his/her sleep on a regular basis. One such example is the Zeo headband, which uses EEG to measure sleep data [28]. The Zeo headband interacts over Bluetooth with a mobile device for data visualization. However, most of these devices though accurate are obtrusive and cumbersome to use, since the user has to wear them and sleep. Logistics such as maintaining bluetooth connection and battery life make these sensors challenging to use. Such headbands might also cause minor discomfort such as numbness, headaches, and skin irritation to the user in the morning. An in-depth survey of 230 participants suggested that, although most people are interested in using technology to track their sleep quality, many are resistant to the idea of having to wear a device during sleep. [14]

Current generation smart phones are equipped with a variety of sensors such as GPS sensors, microphones, image sensors (camera), light sensors, proximity sensors, inertial sensors (accelerometers and gyroscopes), and direction sensors (compass). The small form factor of the smart phones coupled with its ubiquity and the substantial computing power makes them an effective tool for understanding the current state of the smart phone user.

1.2 Aim of the Thesis

To develop and evaluate mathematical models to characterize an individual's sleep pattern by using his/her smartphone accelerometer data. Based on the problem statement mentioned above our work can be divided into following subcategories:

- An android application to collect and save sleep data using inbuilt accelerometer sensor
- Data classification into Sleep/wake using three different algorithms
- Collecting and extracting sleep data from a medically approved embedded EEG sensor
- Using this extracted data to verify the accuracy of the three different algorithms
- Develop a mathematical model to model the sleep of a person. For this we make use of HMM model
- Evaluation of the optimum amount of training data required for HMM model
- An experimental analysis to validate the the accuracy of classification using the algorithm as against existing third party android applications
- Testing the feasibility and robustness of the algorithm in terms of execution on a smartphone

1.3 Organization of the thesis

The rest of the thesis is organized as follows. In chapter 2 we discuss the literature related to the approaches taken to recognize user activity, sleep in specific from a mobile device. Chapter 3 includes a description of our solution approach. It starts with the discussion of general concepts related to modeling of sleep pattern along with an explanation of Hidden Markov Model concepts. This is followed by experimental setup and Data collection. We further discuss various techniques that we explored to classify the obtained data into Sleep/Wake states. Chapter 4 gives the details related to the analysis and results giving a comparison of the performance of the techniques followed by analysis of the technique for modeling of sleep pattern, comparison of obtained HMM parameters, comparison of performance of algorithm with third party android application, performance of our android application (Power consumption) and finally the performance of our algorithm on two different computer architectures. We conclude with Conclusion and Future work as a part of chapter 5 and 6 respectively.

Chapter 2

Related Work

Advances in ubiquitous and pervasive computing have resulted in the development of a number of sensing technologies for capturing information related to human physical activities. Activity recognition is thus mainly a classification problem and the complexity of recognition is activity dependent. For example, detecting running is simpler than that of limb movements during sleep since there is a greater magnitude difference between walking and running when compared to sleeping and awake. The different approaches to activity recognition have been proposed in literature.

Smartphones have been used in the recent past for user activity recognition as mentioned in section 1.1. Such activities are usually defined in the context of the intended application. For instance, an activity recognition system inside a car would try to decipher whether the car is accelerating, decelerating or stopped. Similarly, in the context of a home, previous studies have used smartphones for energy apportionment tasks. Kwapisz et. al. [23] used an Android based smart phone for recognizing very simple activities such as walk, jog, climb up and down the stairs, sit and stand. Yang [30] developed an activity recognition system using the Nokia N95 cell phone for distinguishing between different locomotion. Brezmes et. al. [12] proposed a subject dependent real time activity recognition system again using the Nokia N95 smart phone. Hache et. al. [18] used an accelerometer integrated with a blackberry Bold 9000 platform for detecting changes in the state of the subject caused by starting/stopping and postural changes in activities. Khan et. al. [21] used kernel discriminant analysis for recognizing very simple activities such as walking, up and down the stairs, running and resting on data collected from Samsung Omnia. Zhang et. al. [31] used an HTC smart phone for recognizing again simple activities using a support vector machines.

Approaches to study sleep/wake activity include Polysomnographic studies, environmental sensors and embedded body sensors. Smartphones because of their ubiquitous and non intrusive nature have come up as a emerged as a promising option for sleep monitoring with the help of their inbuilt sensors. Actigraphy has been used to study sleep/wake patterns for the past

20 years. The basic idea is that the state of sleep and wake can be inferred from the amount of body movement during sleep. The advantage of actigraphy over traditional polysomnography (PSG) is that actigraphy can conveniently record continuously for 24-hours a day for days, weeks or even longer [11]. A number of commercial wearable devices have emerged which use embedded accelerometers as a means to monitor sleep. These include the jawbone wrist band [3] and fitbit [4]. Zeo headband uses a combination of inertial sensors and accelerometers. Many of these devices interact with a smartphone over local bluetooth radio for storage, communication with the cloud and data visualization.

Devaul et al. [15] used accelerometers as motion detectors. The purpose of this motion classification system was to make important information about the user's state (whether the user is walking, running, standing still,etc.)available to other applications in real-time. The system consists of a two-layer model that combines a multi-component Gaussian mixture model with Markov models to accurately classify a range of user activity states, including sitting,walking, biking, and riding. Accelerometers have also been used for body-position and posture sensing [17]. Apple's iLife Fall Detection sensor, embeds an accelerometer and a microcomputer to detect falls, shocks, or jerky movements. Active research is being carried out in exploiting this property for determining user context [27]. Lane et al. [24] developed an android application 'BeWell' which sampled the data of three sensors for classification namely : GPS, accelerometer and microphone. Audio sensor data was used to recognize social interaction by classifying (voicing, nonvoicing) audio segments. The accelerometer data was used to classify everyday behaviors necessary to monitor the user's physical activity including (driving, stationary, running, walking). It modeled the sleep duration of the user based on features extracted from the phone, specifically, the frequency and duration of: phone recharging events (since often people recharge their phones overnight); and the periods when the phone is either stationary or in a near silent sound environment. Chen et al. [13] developed a BES (Best Effort Sleep) model which inferred sleep using smartphones in a completely unobtrusive way i.e the user was completely removed from the monitoring process and did not interact with the phone beyond normal user behavior. Features extracted in order to classify consisted of: Light feature, Silence feature, Stationary feature and Phone Usage features. Hao et al. [19] used the built-in microphone of the smartphone to detect the events that were closely related to sleep quality, including body movement, couch and snore, and inferred quantitative measures of sleep quality. They developed a lightweight decision-tree-based algorithm to classify various events based on carefully selected acoustic features, and tracked the dynamic ambient noise characteristics to improve the robustness of classification.

Using multiple sensors in addition to accelerometer results in higher power consumption and a substantial increase in the task complexity. Also these results have not been empirically validated till date. There also have been a number of Android applications in the market for sleep monitoring but there hasn't been much technical analysis regarding their accuracy [5, 6]. Research has been going on in IBM to develop Big Data platform prototype to deliver personalized wellness services for lifestyle management. The open, standards-based platform collects data from mobile wellness sensors including breathing patterns and stress levels, and others to gain insight into an individual's well being [7].

The research prototype collects data from EarlySense's [8] Wellness solution, which uses a contact free sensor placed under any standard mattress. During sleep, the sensors communicate wirelessly with smartphone and tablets to offer information such as sleep analysis, recording heart and breathing rates and patterns. It can also offer trending graphs, as well as an evaluation of restlessness and stress levels for personal use at home. We develop an Android application named SleepSense to collect data using only the accelerometer sensor. We run our algorithm in order to analyze the data collected by this application. We validate our approach empirically by comparing the results obtained with those of Zeo sensor and a third party Android application: Sleep Time [9] from Google Play, which uses accelerometer.

Chapter 3

Solution Approach

We present a novel approach to measure sleep using a mobile device and its inbuilt accelerometer sensor. The accelerometer is able to accurately measure the limb movement, which is a vital parameter to measure sleep quality and quantity. To test the validity of the approach, we compare the analyzed and processed data with that obtained from Zeo, which is a medically approved device for measuring sleep. Zeo uses EEG signal as opposed to our approach based on accelerometer data.

The advantages of using a smartphone for this purpose are:

- Smartphones are ubiquitious
- Non-intrusive

The key principle behind our solution is that limb movements result in movement of the mattress. Since the smartphone is placed on the mattress, its accelerometer can detect movements of the mattress, and in turn that of limb movements. Physiologically, there is hardly any movement when the person is sleeping and comparatively more movement when the person is awake.

We present the implementation of our approach, where data collection is implemented as an Android mobile application and further analysis of the collected data is done in order to measure sleep. We conducted an experimental study with different users using different mobile devices in order to test the robustness of our application. In the experiment, each user recorded the data for twelve consecutive days to maintain uniformity in results. The obtained data was classified into Sleep/Wake states using the techniques described in subsequent sections. The obtained results were compared with those from Zeo, which serves as the ground truth for our approach.

3.1 Modelling of sleep pattern: General Concepts

In probability theory, a purely stochastic system is one whose state is non-deterministic, such that the subsequent state of the system is determined probabilistically. Probabilistic modeling of processes helps to achieve a compact representation of the system characteristics, in terms of probabilities, under varying conditions thus enhancing the ease of comparison. Sleep of a person is an example of such a process because the process is completely different from subject to subject and also shows random day-to-day variations. In order to do a simple qualitative analysis of the sleep of a subject across a number of days, we use a Hidden Markov Model (HMM). The model derived using this can also be used to compare the sleeping pattern of multiple subjects since the very aim of modeling is to characterize an individual's sleep.

In this section, we briefly present HMM and HMM Viterbi Algorithm. The latter is used to take a particular HMM and determine from an observation sequence the most likely sequence of underlying hidden states that might have generated it. We now briefly discuss the concepts behind HMMs.

3.1.1 Hidden Markov Model

A Hidden Markov Model is a statistical Markov model, in which the system being modeled is assumed to be a Markov process [10] with unobserved, i.e. hidden, states. In a HMM, the state is not directly visible, but output dependent on the state, is visible [25]. Each state has a probability distribution over the possible output tokens. Therefore, the sequence of tokens generated by a HMM gives some information about the sequence of states. Some of the key

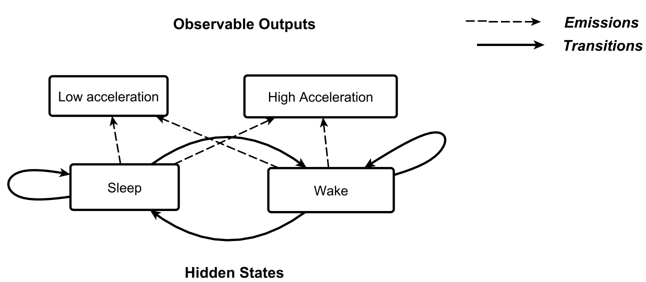


Figure 3.1: HMM state transition diagram of Sleep-Wake process

characteristics of a HMM are:

- Transition probability matrix $T(i, j)$ represents probability of going from state i to j. Order of matrix = $S * S$ where, S is the number of states.

- Emission probability matrix $E(i, k)$ represents probability that output k is emitted from state i. Order of matrix = $(S * V)$ where S = number of hidden states and V = set of vocabulary for observable outputs.

- Posterior Probability matrix $P(i, j)$ is an array with the same length as observation sequence and one row for each state in the model. The (i, j) element of P gives the probability that the model was in state i at the j^{th} step of sequence.

Figure 3.1 shows a 2 state HMM. Here, observable outputs are the acceleration magnitude values obtained from the mobile device. These are grouped into low and high acceleration where as the two hidden states are sleep and wake. The possible state transitions are sleep-sleep, sleep-wake, wake-sleep and wake-wake with each state being associated with an observable output.

3.1.2 Parameter Learning: Expectation Maximization for HMMs

Another question to ask of an HMM is: given a set of observations, what are the values of the state transition probabilities and the output emission probabilities that make the data most likely? These parameters are calculated using the well known Expectation Maximization (EM) algorithm to find the maximum likelihood estimate of the parameters of a hidden Markov model given a set of observed feature vectors.

In statistics, an EM algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step.

3.1.3 HMM Viterbi Algorithm

The HMM Viterbi algorithm deals with the problem of estimating the state sequence which can best represent our observed data. If P is the probability of being at state n at time i, having come from m, T is the transition probability from m to n, and O is the probability of that specific observation at time i from state n. Using Markov property, the current state only depends on the last state, so considering the process from the beginning, probabilities of sequences happening can be calculated by multiplying together the corresponding transition values and observation values with the previous step's probability.

The HMM Viterbi algorithm expands on the idea that at each time step, only the sequence path that has the best probability going into each state needs to be stored. If the model has 2 states, at most 2 paths need to be stored, updated at every time step.

Once the HMM parameters that can explain a set of observation and state sequence are learned using the expectation maximization algorithm, these parameters can be used to find the hidden state sequence for a new set of observation sequence for the same process using the viterbi algorithm.

This concept is applied in our case to learn these parameters for an individual and then use these parameters to classify the data for a different day for the same individual.

3.2 Experimental Setup and Data collection

3.2.1 Experimental Setup

Raw data is collected as a series of instances containing a timestamp and three values corresponding to acceleration along the x-axis, y-axis, and z-axis. Rather than a set sampling rate, the accelerometer in this Android phone triggers an event whenever the accelerometer values change. The rate of events can be set to one of four thresholds: fastest, game, normal, and UI, with fastest being the 'fastest' sampling rate and 'UI' being the slowest. The phones used for this experiment were set to normal. The sampling rate varies because of this but can reach a maximum of 20 Hz. The three axes of acceleration are dependent upon the orientation of the phone. The x-axis runs parallel to the width of the phone, the y-axis runs the length of the phone, and the z-axis runs perpendicular to the face of the phone as shown in figure 3.2. Four

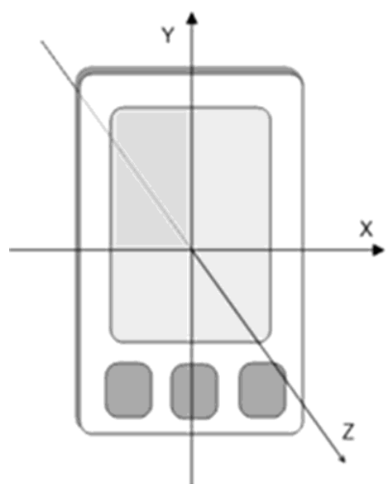


Figure 3.2: Acceleration axes relative to the phone

subjects were asked to put the phone under/near the pillow, while the application was running in order to collect the data. The subject started the application by pressing the start button when going to sleep. It was stopped in the morning after the subject woke up. The sleep data was recorded for this duration. Data was collected for exactly the same duration using three means:

1. Zeo sensor for which the subject tied the Zeo headband on their forehead
2. Our Android data collection app
3. A third-party Android application SleepTime

One phone had the two Android applications running in background. The other phone received the data recorded by Zeo sensor through Bluetooth. This was done so as to compare our application with another accelerometer-based Android application using the same accelerometer sensor chip of the same phone. Zeo which is an altogether different EEG based modality is used

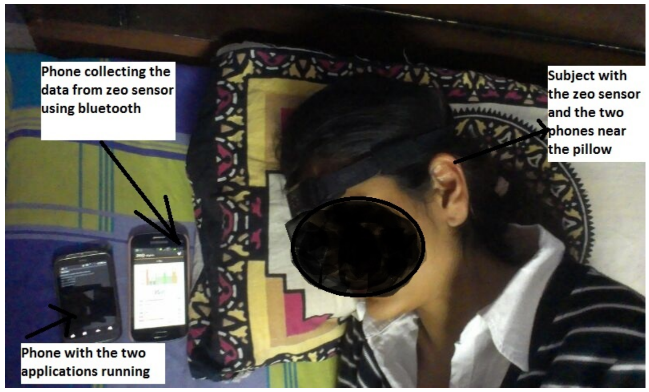


Figure 3.3: Experimental setup. This shows the data collection on two Android phones along with the Zeo sensor. One phone receives the data collected by Zeo using Bluetooth and the other phone had the third party app and our app running.

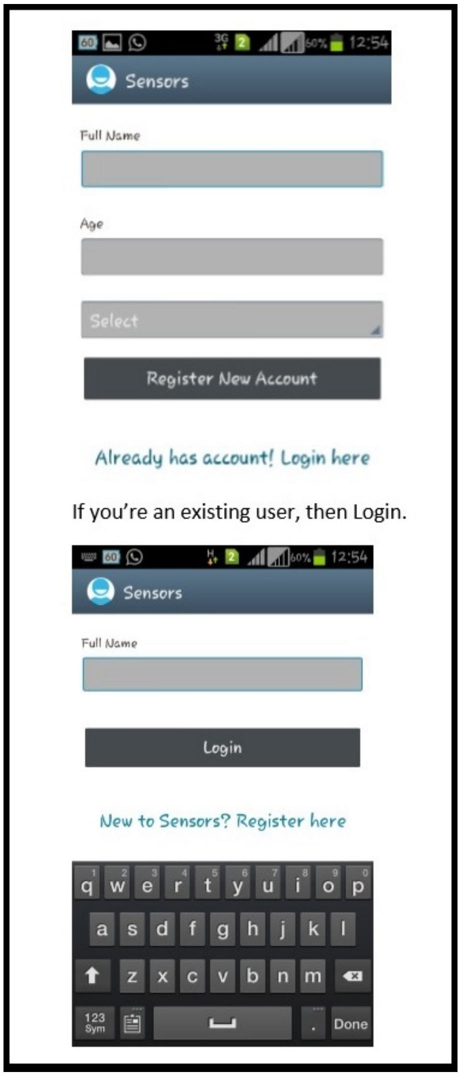
as ground truth to estimate the accuracy of our sleep classification approaches. Figure 3.3 shows the experimental setup with the subject wearing the Zeo headband and data collection going on in two different phones as explained above. Approval of the subject was taken prior to data collection.

3.2.2 Our Android Data Collection Application

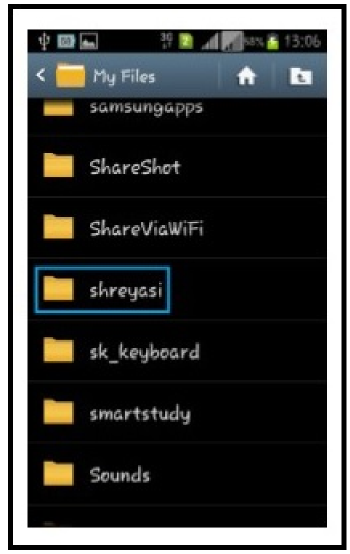
We chose an Android smart phone as the platform for the data collection for multiple reasons: The Android operating system is open source, easily programmable and more importantly, its dominance in the smartphone market. We developed an application to measure sleep and provide the user with an estimate of his/her sleep quality. Once a user registers, a folder with the name of the user will be created on the SD card.The registration and logging in enables multiple users to use the application on the same phone, as different folders will get created for different users. The data would get collected in a CSV file. Users can directly push their data to our institute's centrally hosted server by clicking on the appropriate menu. Figure 3.4a shows the screenshot of the android application with user login screen and Figure 3.4b shows the folder created with user name in the SD card.

3.2.3 Data Collection

Our experimental validation involved two phases. In the calibration phase, we tested for range, resolution and noise in a controlled setting so as to get a fair estimate of the behavior of mobile accelerometers. During this phase data was collected with subjects performing routine activities. This information was used in differentiating between sleep and wake activities in terms of acceleration, which helped in the design of classification methodology.
The testing phase, involved the collection of only sleep data, i.e., subjects started the application before going to sleep and stopped after waking up. Figure 3.5 shows one of the time-acceleration plot for initial data collection spanning over 6 hours. As seen from the figure, the data needs



(a) Login screen



(b) Folder in SD card

Figure 3.4: Our Android application to collect data

processing to detect Sleep/Wake states.

We used different Android phones to establish the generality of our approach including Samsung, HTC-Desire, HTC Wildfire, HTC Sensation, and Sony Xperia. Data from the accelerometer had the following attributes:

- Timestamp
- Acceleration along X, Y, and Z axes
- $Magnitude = \sqrt{x^2 + y^2 + z^2}$

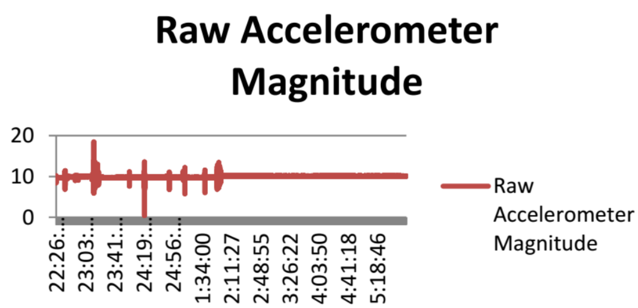


Figure 3.5: Data collection showing raw sleep activity data obtained from accelerometer

3.3 Techniques to Classify Sleep/Wake States

We proceed with our approach keeping in mind that a person can have a minimum of two states of sleep, Sleep and Wake. The raw accelerometer data was processed using three different techniques and results were compared to the ground truth obtained from the Zeo sensor. Zeo classifies data into four states Wake, REM sleep, Light sleep, and Deep sleep as shown in Figure 3.6. Our Sleep/Wake classification was compared to that of Zeo. This was done by mapping Zeo's four states into two. In this mapping, 'Wake' and 'Light' sleep states mapped to 'Wake' whereas 'Deep' sleep and 'REM' sleep mapped to Sleep. From sleep study point of view, this generalization of states is sound. We classify sleep using three techniques. Two of these techniques exploit fixed thresholding for classification where as the third technique uses a probabilistic modeling approach for classification.

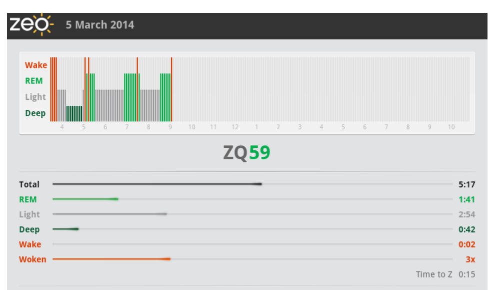


Figure 3.6: Output graph from Zeo sensor classifying sleep into REM, light, deep, wake along with number of times woken and total sleep.

3.3.1 Data extraction from zeo for comparison

Zeo sensor which is based on EEG was used as a ground truth for comparing the performance of all the classification techniques discussed below. Zeo returns a jpg file as an output containing the graph of classification of sleep process into 4 states i.e Wake, REM sleep, Light sleep and Deep Sleep as shown in figure 3.6. Every sample represents a state of sleep for an individual for the sample time duration. The data was extracted by reading the pixel values and every individual colored pixel was assigned a value corresponding to the state. Thus the end result was a vector containing four unique values corresponding to the four states and the length of the vector was equal to the number of sample values.

3.3.2 Classification using Kushida Equation based on Fixed Threshold

The first approach involved classifying the raw accelerometer data using Kushida algorithm [22]. For the raw accelerometer time series data, the acceleration measured at the current time sample was modified according to the accelerometer values of ± 4 time samples. The equation is given below

$$\begin{aligned} A_{modified} = 0.04 * A_{n-4} + 0.04 * A_{n-3} + 0.20 * A_{n-2} \\ + 0.20 * A_{n-1} + 2 * A_n + 0.20 * A_{n+1} + \\ 0.20 * A_{n+2} + 0.04 * A_{n+3} + 0.04 * A_{n+4} \end{aligned} \quad (3.1)$$

where, $A_{modified}$ is the sum of acceleration values at the present time sample and weighted values at the surrounding time samples, A_n is the acceleration value at the current time sample, and $A_{n \pm x}$ are the acceleration values at the surrounding time samples. The algorithm took the modified time series as given in equation 3.1 and the threshold as input and the time series classified into sleep/wake was obtained as the output. This was done to take into account the effect of surrounding epochs on the current epoch sample so as to maintain consistency of results.

If the summed acceleration value obtained from equation 3.1 was found out to be above a certain threshold, the epoch was scored as Wake, otherwise as Sleep. The values of thresholds, used to classify, were varied in three different ranges namely, high, medium, and low.

This involved selecting three threshold values according to the overall nature and magnitude of accelerometer values. This was followed by classification of data. It was observed that a high threshold range gave false sleep epochs and a low threshold gave false wake epochs. Better results were obtained using middle range of thresholds.

3.3.3 Classification using Statistical Method based on Fixed Threshold

The second approach involves a simple statistical and mathematical model in which, the data is first processed by discarding the noisy data in the form of peaks of very high amplitude. The data is then normalized to an amplitude range of 0 to 1. Normalization is done in order to adjust the calibration and other differences of accelerometers of different phones.

This data is viewed in windows of 4 minutes each and one state, i.e., either Sleep or Wake is assigned to every window. The basis for this classification is that if a certain number of samples in each window, in our case 40% of the samples, had magnitude greater than the threshold, that window is classified as Wake otherwise Sleep. The value of the threshold was calculated as follows:

$$Threshold = \frac{Mean + StandardDeviation}{2} \tag{3.2}$$

This Threshold value was calculated separately for every window.

3.3.4 Classification using HMM Training

The final approach involved the classification of raw accelerometer data using Hidden Markov Model training. HMM permits the analysis of non-stationary multivariate time series by modeling the state transition probabilities and the probability of the observation of a state. During the HMM process, the result of the previous state will influence the state recognition result of the next state. This is similar to the process of sleep. Modeling of sleep should consider the relationship between the previous sleep stage and the next sleep stage. As it possesses the properties of successive stage transition, the HMM is a promising model for sleep modeling [26]. This approach involved learning the parameters of HMM namely transition and emission probabilities given a known sequence of observable outputs and hidden states. This was done by the use of expectation-maximization algorithm which is used to give the maximum likelihood estimate of the transition and emission probabilities.

In our case, the inputs were raw accelerometer data down-sampled to match the sampling rate of Zeo sensor and the Zeo sensor output containing two states Sleep and Wake after appropriate mapping as explained in section 3.2.3. 12 days of data was collected for every subject, six to seven hours per day. Transition and emission probabilities were obtained for varying groups of days example 2, 6 and 11 days. It is to be noted that these transition and emission probabilities were estimated using the state map as given by Zeo, which is the ground truth. The obtained probabilities, along with a new set of raw accelerometer data of the same subject, were then used to estimate the sequence of states corresponding to this accelerometer data, i.e., observation sequence.

This involved the use of the viterbi algorithm which calculates the most probable state path of a sequence given the corresponding transition and emission probability parameters for that model. Hence the Sleep/wake states for a new day for the same subject were inferred from the learned probability values.

Chapter 4

Analysis and Results

4.1 Comparison of Accuracy in Classifying Sleep/Wake States

Raw accelerometer data was classified into Sleep/Wake states using three different approaches namely classification using Kushida equation, Statistical method and HMM training as discussed in section 3.3. The result obtained by using each approach was compared to Zeo sensor classification after mapping it to a lower number of states.

The metric used for quantitative comparison is the *Percentage of Matching Samples*. This is calculated as follows. Classification is represented as a row vector, where length of the row vector is total number of time samples and the value of each element was either 1 (for Sleep) or 2 (for Wake).

$$Matching\,Samples = \left(\frac{x}{y}\right) * 100 \tag{4.1}$$

where,
x = Number of matching samples
y = Total number of samples in data

This is a simple and reliable metric for comparison as it captures the accuracy of the classification by direct comparison with the ground truth classified data. This metric comparison can be further extended to give a percentage of false sleep and false wake epochs. Figure 4.1 shows the state maps obtained using Zeo and the other approaches for three different days for a same subject. Comparison of the classification approaches has been summarized in Table 1. The vectors obtained from each approach are compared with that from Zeo and the percentage of the number of matching samples to the total number of samples was found out to be maximum for HMM training approach, greater than 79% on average.

It was also observed that HMM method identified small wake epochs in between sleep epochs more accurately and consistently as compared to the other two approaches.

S.no.	Name of the Approach	Features	Accuracy Provided
1	Kushida Equation approach based on Fixed Threshold	• Data classified using three range of thresholds for each subject. • Different thresholds tried for each range (High, medium and low) • Medium range gave the best results compared to other two • *Threshold fixed before starting the data analysis.*	• Max accuracy 65% • Avg accuracy 59%
2	Statistical Method based on Fixed Threshold	• Noisy peaks removed • Data normalized • Data classified using a simple threshold. • *Threshold derived and then fixed based on the data statistics during the analysis.*	• Max accuracy 74% • Avg accuracy 68%
3	HMM Training	• Noisy peaks removed and data normalized. • Parameters learned and data classified using Expectation maximization and HMM Viterbi respectively with also the help of Zeo sensor. • Obtained probability values used to generate state map for another dataset of same subject • *No fixed threshold. Classification done probabilistically using training data.*	• Max accuracy 84% • Avg accuracy 79%

Table 4.1: Comparison of approaches that classify Sleep/Wake states

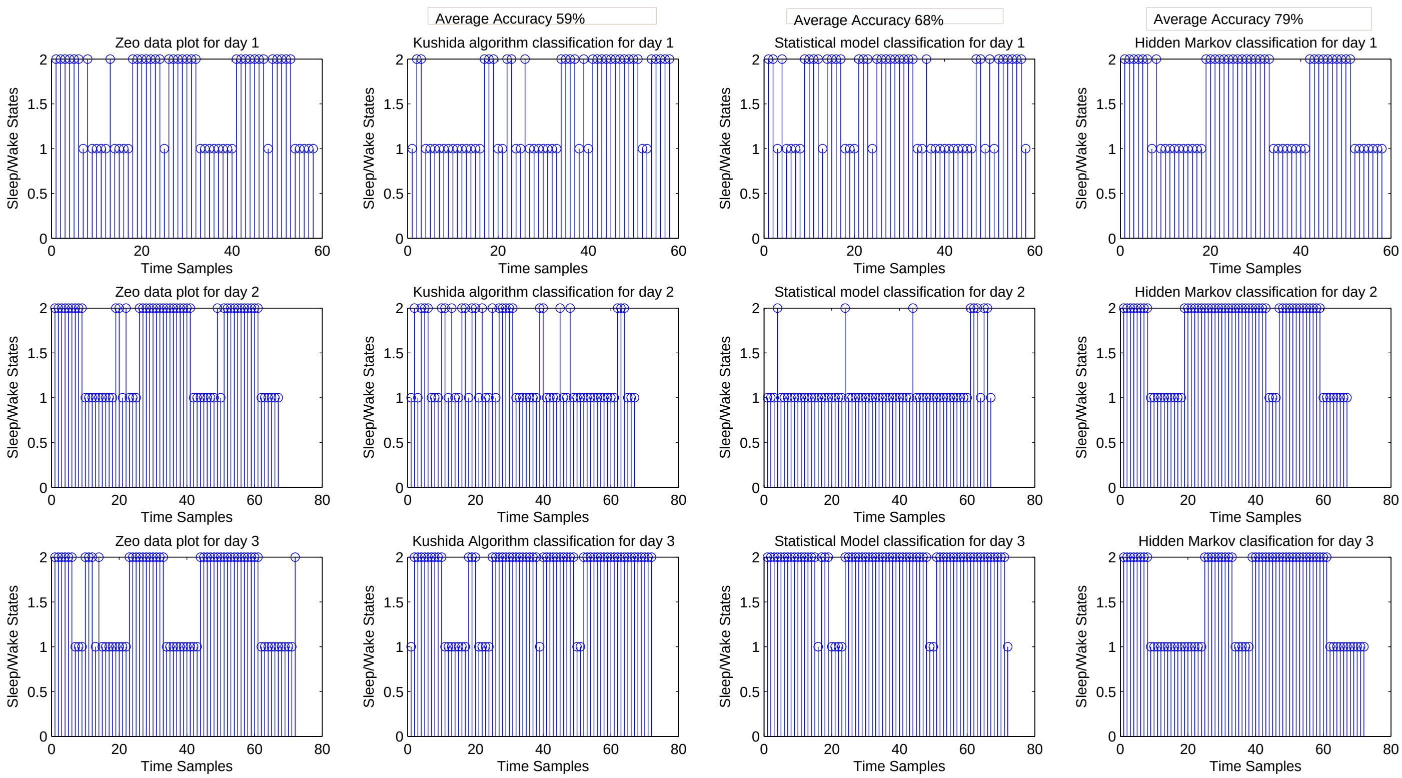


Figure 4.1: Comparison of classified data plots using Zeo, Kushida equation, Statistical method, and HMM training approach for three days. The readings are consistent across the three representative days.

4.2 Analysis of Technique for Modeling of Sleep Pattern

In this section, the focus will be on the HMM training approach which enables classification into Sleep/Wake data as well as modeling of the sleep pattern from HMM parameters such as transition, emission, and posterior probabilities. Classification using HMM technique involved data training using parameter learning and then using the obtained probabilities to find the state sequence for the rest of the days as discussed in section 3.3.4. Data was classified and modeled using varying amount of training data varying from 2 days to 11 days. Figure 4.2 shows the classification as well as posterior probability plot (of wake state) obtained using 2, 6, and 11 days of training respectively.

These plots are for a common day of data so that the results can be compared. The comparison metric was same as that mentioned in section 4.1. It can be seen that the classification is inaccurate as compared to Zeo sensor for 2 days of training data. The classification accuracy for 2 day data was found out to be 69% where as for 6 days it was found out be 84%. A minor improvement of 1% was observed for 11 days of training data in terms of classification. The posterior probability obtained for 6 day training was more in sync with zeo as compared to 2 day training data. Overall, it was observed that the accuracy uniformly increased from 2 to 6 days of training data and after that the increase in percentage accuracy was insignificant. However,

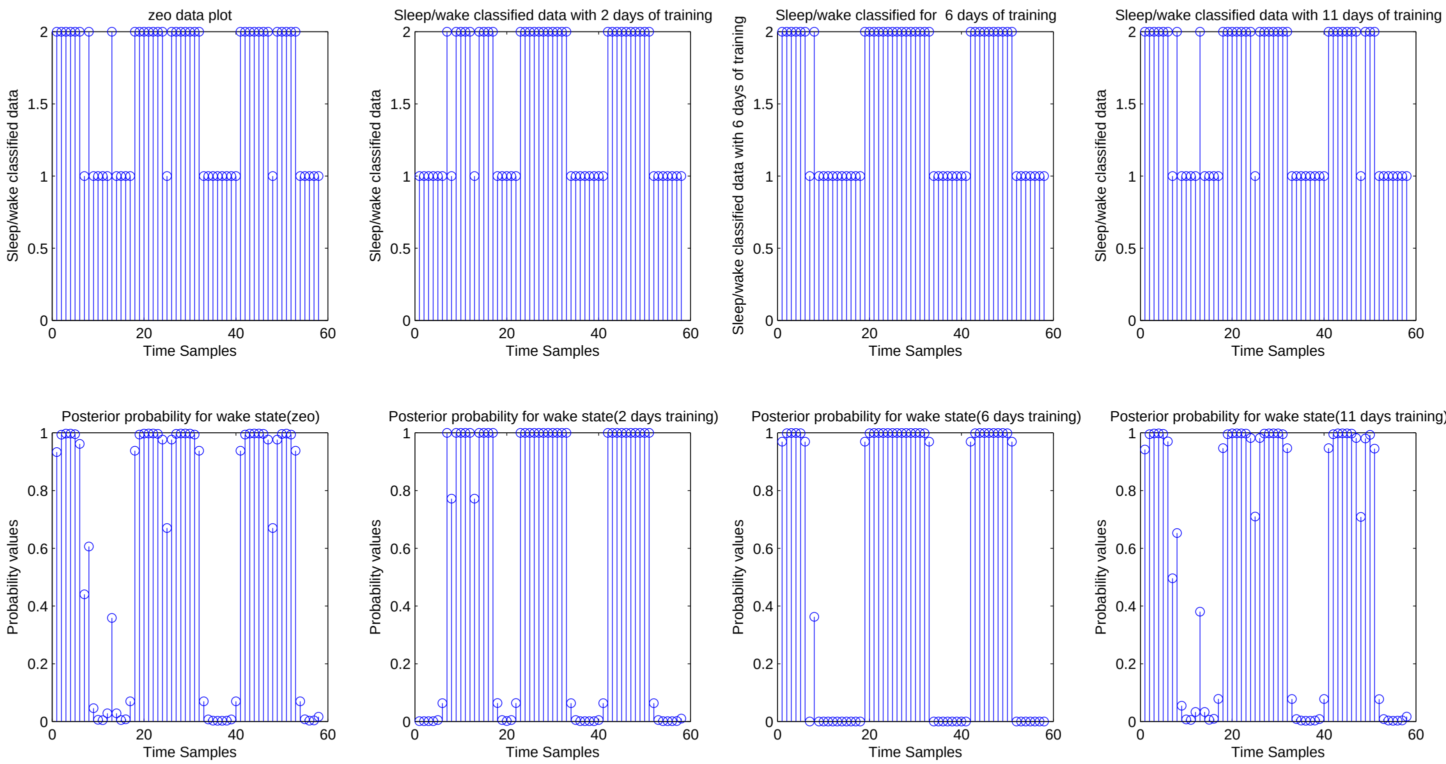


Figure 4.2: Evaluation of HMM approach showing the classification and posterior probability plots for 2, 6, and 11 days of training data respectively. Classification accuracy was 69%, 84% and 85% for 2, 6 and 11 days of training data respectively.

with more training data the model became qualitatively strong as it was observed that it was able to detect wake of small duration in between larger duration of sleep and classify correctly. We conclude that 6 days of training data was fairly sufficient for classification.

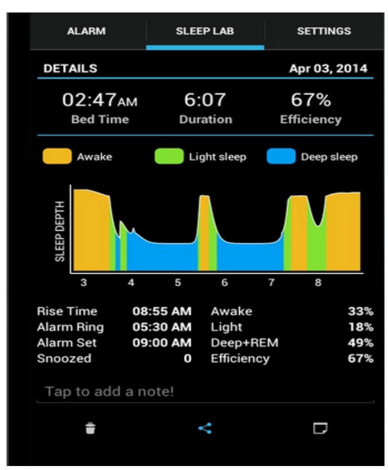


Figure 4.3: Output of the third party Android application : SleepTime

4.3 Comparison of obtained HMM parameters

A pair of Transition and Emission probabilities were obtained for every individual from the Hidden Markov Model after 6 days of training. Transition and emission probabilities were fairly distinct for every individual. The distinctness was more evident in case of subjects belonging to different age groups. Also these parameters were in correlation with the sleep score given by the Zeo. For example, a subject having a higher ZQ score had a higher 'sleep' to 'sleep' and 'wake' to 'sleep' state transition probability compared to a subject with a lower ZQ score. As an example figure 4.4 shows emission probability plots for two subjects belonging to 21-25 years and 40-44 age group respectively. It can be observed that there is a difference between the maximum values of both the plots as well as the average value and also a difference as to how the general pattern varies. Transition probability matrix values for the same subjects were found out to be:

$$\begin{bmatrix} 0.8431 & 0.1569 \\ 0.4380 & 0.5620 \end{bmatrix}$$

$$\begin{bmatrix} 0.7839 & 0.2161 \\ 0.2357 & 0.7643 \end{bmatrix}$$

As mentioned above these probabilities were in sync with ZQ score i.e Subject in lower age group had a higher ZQ and also higher sleep to sleep state transition probabilities as compared to the subject in the higher age group. Obtained parameters can be compared in more detail based on criteria like age group and gender with more of data collection in order to form a more generic training and parametric Hidden Markov Model. This has been included as a part of our future work.

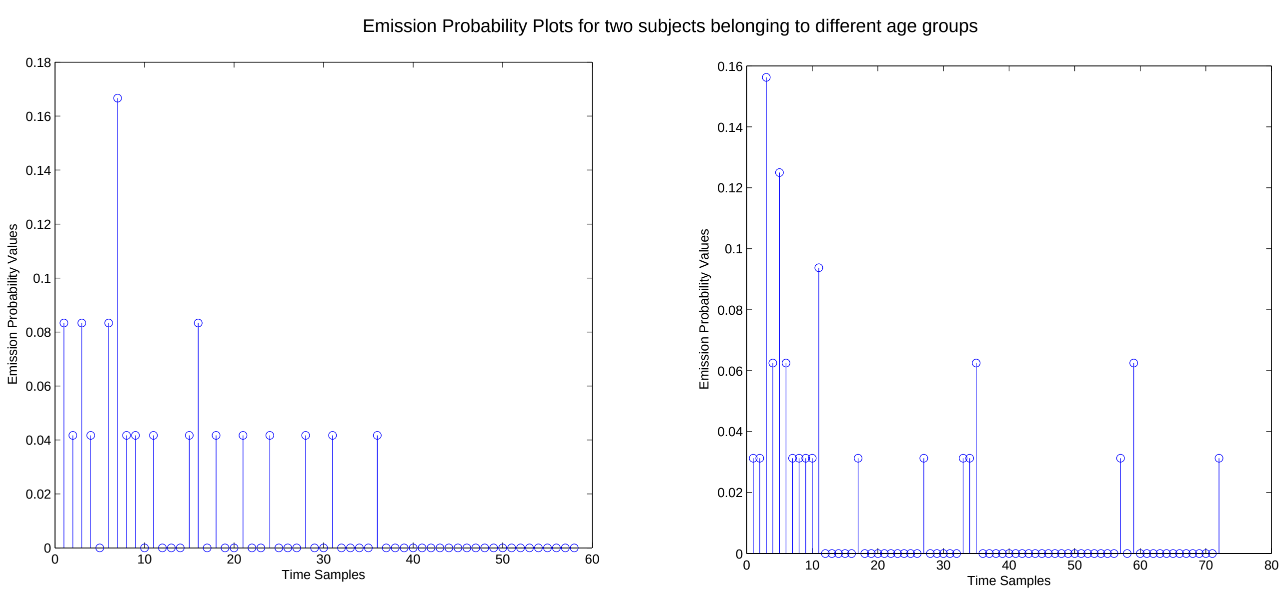


Figure 4.4: Comparison of emission probability obtained from HMM modeling for two subjects of different age groups (a) 21-25 years (b) 40-44 years

4.4 Comparison With Third Party SleepTime Android Application

It can be observed from Table I that the HMM training approach gave better results as compared to the other two methods. In order to prove the robustness of our approach, it was also compared with a third party Android application SleepTime [9]. This application was made to run on the same phone thus collecting data for the same time using the same sensors. It was observed that although the third party Android application classified sleep with a higher level of granularity, i.e., into a higher number of states like Zeo but if the classified states were mapped into Sleep and Wake in the same way as that of Zeo, then our results were closer to that of Zeo. As a sample the percentage of Sleep and Wake after mapping to two states in Figure 4.3 can be observed to be 49% and 51% respectively. For the same data of the same day the categorization using Zeo after mapping was found out to be 68% and 32% respectively. The classification using our HMM training algorithm was 62% and 38% for Sleep/Wake respectively. These comparisons were made using the metric mentioned in 4.1. The observations were consistent, when such analysis was done for a period of 12 days, thus proving the robustness of our algorithm.

4.5 Performance of Our Algorithm

We are working on implementing the detection algorithm on an Android smartphone. In the meantime, we compare performance of our algorithm on two different PC architectures to estimate how our algorithm will perform on an Android smartphone. Table 4.2 mentions execution time of our algorithm on two PC architectures, with more and less resources. The testing was done on HMM training algorithm as it gave the best results during classification. The execution time given in Table 4.2 is for classification and analysis done using 6 days of training data. From the results, it can be inferred that our algorithm is feasible for execution on an Android smartphone, that today has similarly capable resource as S.no. 2 in the table.

S.No.	CPU and RAM	Execution Time
1	Intel core i5 and 6 GB RAM	$\leq$ 1 minute
2	Intel Centrino and 2 GB RAM	$\leq$ 3 minutes

Table 4.2: Performance of our algorithm on two computer architectures

4.6 Performance of our Android Application

Data collection was one of the crucial part of this work. Therefore, evaluating the performance of our application in terms of power consumption during the data collection was necessary. Figure 4.5 shows the power consumption by the application for three different accelrometer sampling rates i.e ‘Fastest’ ,‘Normal’, and ‘UI’ with ‘Fastest’ being the fastest and ‘UI’ being the slowest. The power consumption was measured according to how much time was taken for the battery to drop from 100 percent to 60 percent (approx). Also the baseline consumption of the phone

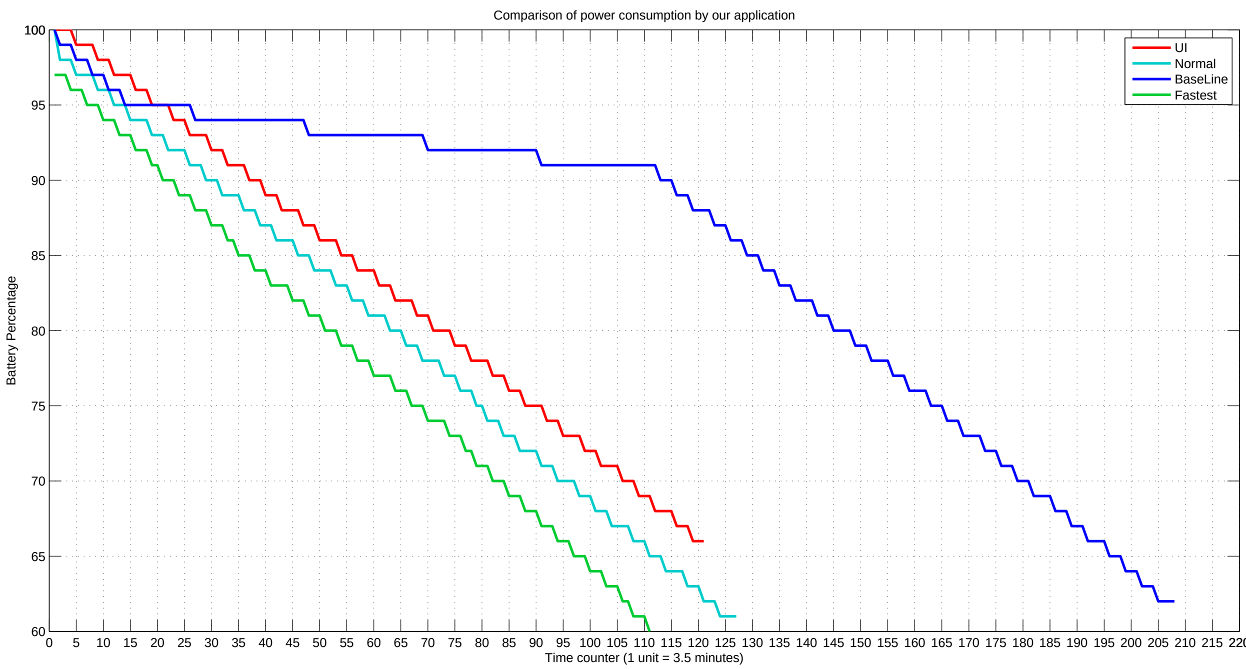


Figure 4.5: Comparison of power consumption against different accelerometer sampling rates along with baseline consumption of the phone

when accelerometer was not doing any sampling was also measured. The operating conditions were kept exactly same for all the four cases. From the figure it can be observed that ’UI’ sampling rate took the longest time to consume 40 percent battery followed by normal and then fastest with fastest taking the least time. Thus It is clear that normal sampling rate which we used was fair enough to give a long duration data collection with sufficient number of samples. The data collection frequency collection can also be increased to ’Fastest’ which can also give a continuous data collection of upto 7-8 hours based on this plot. The screen of the phone was kept on for the whole time. This was done so as to evaluate the performance of the application in worst conditions. Even longer duration data collection and better performance can be achieved by turning off the screen during data collection. Therefore the data collection accuracy can be improved by switching to ’Fastest’ mode which will be able to give more samples without much battery consumption as compared to ’Normal’ mode.

Chapter 5

Conclusion

In this work, the problem of detecting and classify sleep using the accelerometer data from a smartphone was first considered. This involved the making of an android application to collect accelerometer data in the explained experimental setup. Three approaches were examined to classify data where two were based on fixed threshold and the third approach was based on probabilistic learning. It was observed that the probabilistic learning approach using HMM gave the best results among the three, when compared with Zeo. After classification the classified data was modeled using HMM and transition, emission and posterior probabilities were calculated. It was also observed that the results showed remarkable uniform improvement initially with increase in training data. Little quantitative improvement was observed for more than 6 days of training data. Although large amount of training data improved the overall capability to detect small duration wake epochs, fairly accurate results were obtained with less training of data. The performance of classification using HMM was also compared with a third party android application. Performance of this algorithm was also compared on two different computer architectures to estimate the feasibility of execution on an android smartphone.

Chapter 6

Future work

In present work, sleep cycles were identified using a number of approaches and sleep was modeled using a simple first order HMM. This model can be extended to classifying sleep into more number of states by increasing the order of the model. In our future work, we will evolve our classification into all four states as that of Zeo.
Also the current analysis was done offline. We plan to integrate this analysis and do it online as a part of an android mobile application where the user will be able to see and visualize the results of his/her sleep monitoring as a part of the application on the phone. The results can also be analysed further to draw some interesting inferences. These may include : Analysis and comparison of results based on subject demographics example old vs young, male vs female which can be used to establish if different models are needed for different groups of user. We further plan to compare the results with other machine learning techniques This system can also be further extended using auxillary sensors. For example, sleep apnea [2, 29] can be detected by using the mobile accelerometer sensor with an additional external pulse oximeter sensor. This sensor measures the oxygen saturation and pulse rate of a subject. These readings along with the data of body movement can be used to detect the probability that a person might be suffering from sleep apnea.

Bibliography

[1] http://en.wikipedia.org/wiki/Sleep_disorder.

[2] http://en.wikipedia.org/wiki/Sleep_apnea.

[3] https://jawbone.com/up.

[4] http://www.fitbit.com/.

[5] http://www.sleepcycle.com/.

[6] https://sites.google.com/site/sleepasandroid/.

[7] http://www-03.ibm.com/press/uk/en/pressrelease/43425.wss.

[8] http://www.earlysense.com/.

[9] https://play.google.com/store/apps/details?id=com.azumio.android.sleeptime&hl=en.

[10] http://en.wikipedia.org/wiki/Markov_process.

[11] Ancoli-Israel, S., Cole, R., Alessi, C., Chambers, M., Moorcroft, W., and Pollak, C. The role of actigraphy in the study of sleep and circadian rhythms. american academy of sleep medicine review paper. *Sleep 26*, 3 (2003), 342–392.

[12] Brezmes, T., Gorricho, J.-L., and Cotrina, J. Activity recognition from accelerometer data on a mobile phone. In *Distributed computing, artificial intelligence, bioinformatics, soft computing, and ambient assisted living*. Springer, 2009, pp. 796–799.

[13] Chen, Z., Lin, M., Chen, F., Lane, N. D., Cardone, G., Wang, R., Li, T., Chen, Y., Choudhury, T., and Campbell, A. T. Unobtrusive sleep monitoring using smartphones. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on* (2013), IEEE, pp. 145–152.

[14] Choe, E. K., Consolvo, S., Watson, N. F., and Kientz, J. A. Opportunities for computing technologies to support healthy sleep behaviors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2011), ACM, pp. 3053–3062.

[15] DeVaul, R. W., and Dunn, S. Real-time motion classification for wearable computing applications. *2001, project paper, http://www. media. mit. edu/wearables/mithril/realtime. pdf* (2001).

[16] Estévez, P., Held, C., Holzmann, C., Perez, C., Pérez, J., Heiss, J., Garrido, M., and Peirano, P. Polysomnographic pattern recognition for automated classification of sleep-waking states in infants. *Medical and Biological Engineering and Computing 40*, 1 (2002), 105–113.

[17] Foerster, F., Smeja, M., and Fahrenberg, J. Detection of posture and motion by accelerometry: a validation study in ambulatory monitoring. *Computers in Human Behavior 15*, 5 (1999), 571–583.

[18] Hache, G., Lemaire, E., and Baddour, N. Mobility change-of-state detection using a smartphone-based approach. In *Medical Measurements and Applications Proceedings (MeMeA), 2010 IEEE International Workshop on* (2010), IEEE, pp. 43–46.

[19] Hao, T., Xing, G., and Zhou, G. isleep: unobtrusive sleep quality monitoring using smartphones. In *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems* (2013), ACM, p. 4.

[20] Jean-Louis, G., Kripke, D. F., Mason, W. J., Elliott, J. A., and Youngstedt, S. D. Sleep estimation from wrist movement quantified by different actigraphic modalities. *Journal of neuroscience methods 105*, 2 (2001), 185–191.

[21] Khan, A. M., Lee, Y.-K., Lee, S., and Kim, T.-S. Human activity recognition via an accelerometer-enabled-smartphone using kernel discriminant analysis. In *Future Information Technology (FutureTech), 2010 5th International Conference on* (2010), IEEE, pp. 1–6.

[22] Kushida, C. A., Chang, A., Gadkary, C., Guilleminault, C., Carrillo, O., and Dement, W. C. Comparison of actigraphic, polysomnographic, and subjective assessment of sleep parameters in sleep-disordered patients. *Sleep medicine 2*, 5 (2001), 389–396.

[23] Kwapisz, J. R., Weiss, G. M., and Moore, S. A. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter 12*, 2 (2011), 74–82.

[24] Lane, N. D., Mohammod, M., Lin, M., Yang, X., Lu, H., Ali, S., Doryab, A., Berke, E., Choudhury, T., and Campbell, A. Bewell: A smartphone application to monitor, model and promote wellbeing. In *5th International ICST Conference on Pervasive Computing Technologies for Healthcare* (2011), pp. 23–26.

[25] Leroux, B. G. Maximum-likelihood estimation for hidden markov models. *Stochastic processes and their applications 40*, 1 (1992), 127–143.

[26] Pan, S.-T., Kuo, C.-E., Zeng, J.-H., Liang, S.-F., et al. A transition-constrained discrete hidden markov model for automatic sleep staging. *Biomed. Eng. Online 11* (2012), 52.

[27] Randell, C., and Muller, H. Context awareness by analysing accelerometer data. In *Wearable Computers, The Fourth International Symposium on* (2000), IEEE, pp. 175–176.

[28] Rubin, B. Multi-modal sleep system, Sept. 6 2011. US Patent App. 13/226,121.

[29] Williams, A. J., Yu, G., Santiago, S., and Stein, M. Screening for sleep apnea using pulse oximetry and a clinical score. *CHEST Journal 100*, 3 (1991), 631–635.

[30] Yang, J. Toward physical activity diary: motion recognition using simple acceleration features with mobile phones. In *Proceedings of the 1st international workshop on Interactive multimedia for consumer electronics* (2009), ACM, pp. 1–10.

[31] Zhang, S., McCullagh, P., Nugent, C., and Zheng, H. Activity monitoring using a smart phone's accelerometer with hierarchical classification. In *Intelligent Environments (IE), 2010 Sixth International Conference on* (2010), IEEE, pp. 158–163.