AN UNSUPERVISED METHOD FOR WAKE/SLEEP SCORING

by

JIN XING

Submitted in partial fulfillment of the requirements

For the degree of Master of Science

Thesis Advisor: Dr. Kenneth Loparo

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

January, 2014

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis/dissertation of

_____

candidate for the _____degree *.

(signed)_____
            (chair of the committee)

_____

_____

_____

_____

_____

(date) _____
            Nov. 26, 2013

*We also certify that written approval has been obtained for any
proprietary material contained therein.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AR   Autoregression

D2   Correlation Dimension

DHMM   Discrete Hidden Markov Model

EEG   Electroencephalography

EMD   Empirical Model Decomposition

EMG   Electromyography

EOG   Electrooculography

FCM   Fuzzy C-Means

FFT   Fast Fourier Transform

GK-FCM   Gustafson–Kessel algorithm

k-NN   k-Nearest Neighbor

LZ   Lempel-Ziv

NREM   Non-Rapid Eye Movement

PCA   Principal Component Analysis

PSD   Power Spectral Density

PSG   Polysomnography

REM   Rapid Eye Movement

RSP   Relative Spectral Power

SampEn   Sample Entropy

SE   Spectral Entropy

SEF   Spectral Edge Frequency

SVD   Singular Value of Decomposition

SVM   Support Vector Machine

An Unsupervised Method for Wake/Sleep Scoring

Abstract

by

JIN XING

Visual sleep scoring of Polysomnograms (PSG) by an expert is a time-consuming process. Although a number of automatic sleep scoring methods have been proposed in literature, most of them are based on supervised algorithms, that is, labels in their training data assigned by an expert are required. In this thesis, we propose an unsupervised method for wake/sleep scoring without labels a priori. Features based on temporal and spectral analysis are extracted from a single channel of EEG. Principal Component Analysis (PCA) is used to reduce the number of features while identifying patterns in the data. The Gustafson–Kessel algorithm is used for clustering analysis and sleep scoring is done by retrieving one characteristic feature of wake: the alpha rhythm. Sixteen subjects from the MIT-BIH Polysomnographic Database were tested by this method. Compared to actual stage scoring, 14 have scoring accuracy above 75% and the average accuracy is 79.35%.

1.   INTRODUCTION

Polysomnography (PSG) is a comprehensive recording of the biophysiological changes that occur during sleep. It consists of various electrical biosignals including electroencephalography (EEG), electrooculography (EOG) and electromyography (EMG). These signals are segmented into epochs of 30 seconds and assigned a sleep stage by an expert. This procedure is called sleep scoring. However, it is a time consuming and subjective process. Thus, the development of an automatic sleep scoring system is desirable.

Automatic sleep scoring has been addressed by many research groups. Supervised methods we discussed first. In [15], Hugo Simões et al. used R-square Pearson correlation coefficient and Bayesian classifier; using 19 of the most discriminate features selected from 204 features in 6-channel EEG, a 93% agreement with the expert is obtained for 2-class (wake and sleep) detection. In [31], Zhou Peng, et al (2011) combines Principal Component Analysis (PCA) and Support Vector Machine (SVM) to discriminate stage W, NREM stage 2 and NREM stage 3 for each subject; by testing 5 subjects, 87.9% accuracy is achieved on average. [13] (Salih Gunes et al, 2010) proposed a feature weighting method based on k-means clustering and combined it with k-NN (k-nearest neighbor) and decision tree classifiers to classify sleep EEG into six sleep stages; it achieved 82.15% success rate using k-NN classifier for k a value of 40. Farideh Ebrahimi et al (2008) in [8], deployed wavelet packet

coefficients and artificial neural network and their result demonstrated that 4 sleep stages could be automatically discriminated with an accuracy of $93.0 \pm 4.0\%$. In [27] Shing-Tai Pan et al (2012) developed a classification system based on Discrete Hidden Markov Model (DHMM) and 85.29% overall agreement between the expert and the results is presented. Other then supervised learning based methods, a few unsupervised sleep scoring schemes are also proposed. In [30], Hese et al (2001) implemented a semi-automatic method based on a modified version of k-means algorithm. In [11], I. Gath et al (1989) suggested computerized scoring of sleep EEG into various stages by fuzzy clustering. However, no result is presented in these two works. Jing Dong et al (2010) in [7], applied Empirical Model Decomposition (EMD) and k-means algorithm to stage wakefulness and three NREM sleep stages; only $60 \pm 5.0\%$ agreement with the expert was attained. All of the high performances reported for their methods are based on supervised learning, that is, training data labeled by an expert is required. Therefore, it's worthwhile to investigate the design of an unsupervised sleep scoring method without any label a priori. Such a system makes it possible to monitor a patient without requiring a sleep scoring of the PSG by an expert. Also, it would increase the time efficiency and reproducibility of sleep scoring. To this end, this thesis proposes an unsupervised wake/sleep scoring method, as the first step toward developing a complete sleep stages unsupervised scoring system. Our method only requires a single EEG channel. The similar patterns in EEG presented between wake and NREM (non-rapid eye movement) stage 1 or REM (rapid eye

movement) is what makes automated wake/sleep scoring challenging. Although other biosignals, such as EOG and EMG, maybe helpful, they are not used in our study. The data used in this study is from MIT-BIH Polysomnographic Database. All the algorithms are developed in the MATLAB environment.

The thesis is organized as follows: Chapter 2 reviews the background algorithms used in our method; Chapter 3 gives the schematic overview of our method and discusses each of its steps in detail; Chapter 4 presents the results; Chapter 5 contains conclusions and future work.

2. BACKGROUND ALGORITHMS

2.1 Welch's Method

A method to estimate the power spectrum of a signal is to find the discrete-time Fourier transform of the signal and compute the magnitude squared of the result. This estimate is called the periodogram. For a signal $\{x(n)\}_{n=1}^{N}$, the periodogram spectral estimator is computed as follows:

$$\hat{P}(f) = \frac{1}{N}\left|\sum_{n=1}^{N} x(n)\exp(-j2\pi fn)\right|^2 \tag{2.1.1}$$

where $\hat{P}(f)$ is the estimation of the periodogram. To ensure the estimate is asymptotically unbiased, that is, as the number of samples increases, the expected value of the periodogram approaches the true power spectral density (PSD), the magnitude squared of the FFT is scaled by the signal length N. The problem with the periodogram estimate of the PSD is that its variance is large and does not decrease as the number of samples increases.

Welch's method is a modified periodogram, where the data segments are windowed before computing the periodogram. In Welch's method, signals are divided into overlapping segments, each data segment is windowed, periodograms are calculated and the average of the periodograms is computed. $x_l(n), l = 1,...,S$ are data segments and each segment's length equals M. The overlap ratio is frequently chosen as 50% ($M/2$). The Welch PSD estimate is given by:

4

$$\hat{P}_w(f) = \frac{1}{S}\sum_{l=1}^{S}\hat{P}_l(f) \tag{2.1.2}$$

$$\hat{P}_l(f) = \frac{1}{M}\frac{1}{P}\left|\sum_{n=1}^{M}v(n)x_l(n)\exp(-j2\pi fn)\right|^2 \tag{2.1.3}$$

where $\hat{P}_w(f)$ is the Welch PSD estimate, S is the number of segments, $\hat{P}_l(f)$ is the periodogram estimate of the l th segment, $v(n)$ is the data-window, M is the length of window sequence and of each signal segment. P is the total average of $|v(n)|^2$ and given as $P = 1/M\sum_{n=1}^{M}|v(n)|^2$. Since the spectrum of a finite-length signal typically exhibits side-lobes due to discontinuities at the endpoints, a nonrectangular data window is applied to reduce the spread of the spread of the spectral energy into the side-lobes of the spectrum or "spectral leakage" while increasing the width of spectral peaks. Moreover, with a suitable window (such as Hamming, Hanning, or Kaiser), overlap rates of about half the section length have been found to lower the variance of the estimate significantly. Welch has shown that, for half-overlap:

$$\mathrm{var}(\hat{P}_w(f)) = \frac{9}{8S}\mathrm{var}(\hat{P}_l(f)) \tag{2.1.4}$$

2.2   Harmonic Parameters

The Harmonic parameters used are the center frequency (f_c), the bandwidth (f_σ) and the spectral value at the center frequency (S_{f_c}), defined as follows:

$$f_c = \sum_{f_L}^{f_H} f P_{xx}(f) \bigg/ \sum_{f_L}^{f_H} P_{xx}(f) \qquad (2.2.1)$$

$$f_\sigma = \left(\sum_{f_L}^{f_H} (f - f_c)^2 P_{xx}(f) \bigg/ \sum_{f_L}^{f_H} P_{xx}(f) \right)^{1/2} \qquad (2.2.2)$$

$$S_{f_c} = P_{xx}(f_c) \qquad (2.2.3)$$

where, $P_{xx}(f)$ is the spectral density function, which is calculated for the frequency band $\{f_L, f_H\}$.

2.3   Spectral Edge Frequency

The spectral edge frequency (SEF), usually expressed as "SEF p", is defined as:

$$\sum_{f_L}^{SEF} P_{xx}(f) = p \sum_{f_L}^{f_H} P_{xx}(f) \qquad (2.3.1)$$

It stands for the frequency, up to which p percent of the total power of the frequency band $\{f_L, f_H\}$ is accumulated.

2.4   Spectral Entropy

The spectral entropy (SE) is given by:

$$SE = -\sum_{f_L}^{f_H} S_{xx}(f) \log S_{xx}(f) \bigg/ \log(no.\,of\ discrete\ frequencies) \qquad (2.4.1)$$

where $S_{xx}(f)$ is relative power, defined as:

$$S_{xx}(f) = P_{xx}(f) \bigg/ \sum_{f_L}^{f_H} P_{xx}(f) \tag{2.4.2}$$

i.e. percentage of the total power. By dividing through by the logarithm of the number of discrete frequencies, the spectral entropy can be normalized on the interval [0, 1].

2.5 Autoregressive Model

Autoregressive (AR) models of order p are defined by:

$$x_t = -\sum_{k=1}^{p} a_k x_{t-k} + y_t \tag{2.5.1}$$

where x_t is the signal at time t, a_k are coefficients of the AR model and y_t is a zero mean white noise signal and the signal at time t is a linear combination of the past p signals plus a white noise. If the AR model is being fit to measured data, the AR coefficients a_k minimize the mean-square prediction error of the model.

2.6 Hjorth Parameters

Hjorth parameters are computed from the variance of the signal x and its the first and second derivatives x', x''. If we denote the variance of x as $\mathrm{var}(x)$, then the Hjorth parameters are defined as follows:

$$Activity = \sqrt{\mathrm{var}(x)} \tag{2.6.1}$$

7

$$Mobility = \sqrt{\text{var}(x')/\text{var}(x)}$$

(2.6.2)

$$Complexity = \sqrt{\text{var}(x'') \times \text{var}(x)/\text{var}(x')^2}$$

(2.6.3)

The activity is the signal standard deviation. The mobility measures the spread of the changes in the signal compared to the spread of the signal. The complexity is a measure of how complicated the signal is.

2.7 Sample Entropy

Sample entropy (SampEn) is a measure of predictability/ regularity of a time series and assigns a non-negative number to the sequence, with larger values corresponding to more irregularity in the data. SampEn is the negative natural logarithm of the conditional probability that two sequences similar for m points remain similar for $m+1$ points, within a tolerance r, excluding self-matches. Thus, for a time series of N points, $\{x(n), n = 1,...,N\}$, the $k = 1,...,N-m+1$ vectors of pattern length m are formed as $X_m(k) = \{x(k+i), i = 0,...,m-1\}$. The distances among vectors are calculated as the maximum absolute distance between their corresponding scalar elements. The number of within vectors distance r of the vector i is denoted by B_i. Counting the number of different vectors and normalizing yields:

$$B^m(r) = \frac{1}{N-m} \sum_{i=1}^{N-m} \frac{B_i}{N-m+1}$$

(2.7.1)

Repeating the process for vectors of pattern length $m+1$, $B^{m+1}(r)$ can be calculated and SampEn is computed as:

$$SampEn(m, r, N) = -\ln\left[\frac{B^{m+1}(r)}{B^m(r)}\right] \qquad (2.7.2)$$

Various applications have shown that $m=1$ or 2 are reasonable choices, and $r = 0.1 \sim 0.2SD$ are suitable values to use. In this study, we take parameters $m=2$ and $r = 0.2SD$.

2.8 Correlation Dimension

Correlation dimension (D2) is a measure that quantifies the complexity of the signal by estimating the dimension of the attractor. It is estimated according to the Grassberger and Procaccia algorithm:

$$D2 = \lim_{r \to 0}\left[\frac{\log_2(C_m(r))}{\log_2(r)}\right] \qquad (2.8.1)$$

where m is the embedding dimension; $C_m(r)$ is the correlation integral defined as:

$$C_m(r) = \frac{1}{N_p}\sum_{i,j=1;i<j}^{N_p}\Theta\left(r - \left\|y_i - y_j\right\|\right) \qquad (2.8.2)$$

where the Heaviside function $\Theta(x) = 1$ for $x > 0$, $\Theta(x) = 0$ otherwise; y_i are state vectors in the embedding space; r is radius (starts at 0 and ends at the maximum distance of the y_i pairs); N_p is the number of different y_i pairs. The

9

value of D2 is calculated from the slope within the scaling region of the curves $\log C_m(r)$ versus $\log(r)$. A more detailed description of correlation dimension, from a mathematical point of view, can be found in [9, 18].

2.9 Lempel-Ziv Complexity

Lempel-Ziv (LZ) complexity has been applied extensively in biomedical signal analysis to estimate the complexity of discrete-time physiologic signals. Before calculating the LZ complexity $c(n)$, typically the discrete-time biomedical signal $x(n)$ is converted into a binary sequence. By comparison with the threshold T_d, usually the median, the signal is converted into a 0-1 sequence P as follows:

$$P = s_1, s_2, ..., s_n \qquad (2.9.1)$$

Where

$$s_i = \begin{cases} 0, & if \ x(i) < T_d \\ 1, & otherwise \end{cases} \qquad (2.9.2)$$

To compute LZ complexity, the sequence P is scanned from left to right and the complexity counter $c(n)$ is increased by 1 every time a new subsequence of consecutive characters is encountered. The LZ complexity is calculated as follows:

Let S and Q denote two subsequences of P and SQ be the concatenation of S and Q, while sequence SQv is derived form SQ after its last character is deleted

(v denotes the operation of deleting the last character in the sequence). Let $V(SQv)$

denote the vocabulary of all different subsequences of SQv. Initially,

$c(n) = 1$, $S = s_1$, $Q = s_2$, hence, $SQv = s_1$. In general, $S = s_1, s_2, .., s_r, Q = s_{r+1}$ and

$SQv = s_1, s_2, ..., s_r$. If $Q \in V(SQv)$ (i.e., Q is a subsequence of SQv), renew Q to

be s_{r+1}, s_{r+2} and judge once again if $Q \in V(SQv)$ or not, and so forth until

$Q \notin V(SQv)$ (i.e., Q is a new sequence), then $Q = s_{r+1} s_{r+2}, ..., s_{r+i}$ is not a

subsequence of $SQv = s_1, s_2, ..., s_{r+i-1}$, increase $c(n)$ by 1 and then renew

$S = s_1, s_2, ..., s_{r+i}, Q = s_{r+i+1}$. Repeat the above procedure until Q is the last character

of P. Then, $c(n)$, the number of different patterns in P , is the LZ complexity.

According to Lempel and Ziv, for a 0-1 sequence:

$$\lim_{n \to \infty} c(n) = b(n) = n/\log_2 n \qquad (2.9.3)$$

In order to obtain the complexity measure which is independent of the sequence

length, $c(n)$ is normalized to compute the normalized LZ complexity:

$$C(n) = c(n)/b(n) \qquad (2.9.4)$$

The normalized LZ complexity reflects the rate of new patterns in a given sequence.

$C(n) = 0$ implies an ordered process, $C(n) = 1$ a totally stochastic process. The

larger the value of $C(n)$, the closer the sequence is to a stochastic process.

2.10   Principal Component Analysis (PCA)

PCA is a linear transformation that projects a data set onto a set of orthogonal coordinates, i.e. the eigenvectors of its covariance matrix. In this way, it highlights the similarities and differences contained in the data. Choosing the coordinates with the highest variances, i.e. those corresponding to the largest singular values of the covariance matrix, while eliminating coordinates with low variance, the dimension of the data can be reduced. In principal component space the data is expressed in a way that is easier for identifying patterns, while retaining most of the information in the original space.

Let X be a $n \times N$ matrix of real numbers, $X \in \mathbb{R}^{n \times N}$, where n is the number of features and N is the number of epochs. The column i of X represents the feature vector of the i th epoch. The correlation matrix of X is given by:

$$C = \frac{1}{N} X \cdot X^T \qquad\qquad (2.10.1)$$

where we have removed the mean of the data: $X_{i,j} = X_{i,j} - \bar{X}_i$. Since C is real symmetric ($n \times n$) matrix, it has a singular value decomposition (SVD) of the form: $C = U \cdot \Lambda \cdot U^T$, where U is an orthogonal matrix (matrix of orthogonal unit vectors: $U^T U = I$) and Λ is a diagonal matrix of singular values. The columns of U are the singular vectors of C and the diagonal elements of Λ are the corresponding singular values. If $\Lambda = diag\left\{\lambda_1, \lambda_2, ..., \lambda_n\right\}$, $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n$, correspondingly the singular vectors are sorted in order of significance, then the PCA transformation of

X is given by:

$$S = U^T \cdot X \tag{2.10.2}$$

which is the projection of the original feature matrix onto the orthogonal coordinates.
The singular vectors of C are called the principal components. By selecting only the
first d rows of S, we have reduced the feature vectors in X from n to d. This
is the dimensional reduction characteristic of PCA.

2.11   The Gustafson–Kessel Algorithm (GK-FCM)

The Gustafson–Kessel algorithm extends the standard fuzzy c-means (FCM)
algorithm by including an augmented version of the Euclidean distance to be in the
form:

$$D_{ikA_i}^2 = \left(x_k - v_i \right)^T A_i \left(x_k - v_i \right), \ 1 \leq i \leq c, \ 1 \leq k \leq N \tag{2.11.1}$$

The matrices A_i serve as optimization variables in the c-means function, allowing
each cluster to adapt the distance norm to the local topological structure of the data.
The Gustafson–Kessel algorithm minimizes the following objective function:

$$J(X;U,V,A) = \sum_{i=1}^{c} \sum_{k=1}^{N} \left(\mu_{ik} \right)^m D_{ikA_i}^2 \tag{2.11.2}$$

where, $U = \left[\mu_{ik} \right]$ is a fuzzy partition matrix of the data $X \in \mathbb{R}^{n \times N}$, c is the number
of clusters, $V = \left[v_1, v_2, ..., v_c \right], v_i \in \mathbb{R}^n$ is a c-tuple of cluster prototypes and the

13

weighting parameter $m \in (1, \infty)$ determines the fuzziness of the resulting clusters (as $m \rightarrow 1$, the partition becomes hard; $m \rightarrow \infty$, the partition becomes maximally fuzzy). The optimization of the objective function is subject to the standard constraints:

$$\forall i, k \ u_{ik} \in [0,1], \ \forall i \ 0 < \sum\nolimits_{k=1}^{N} u_{ik} < N, \ and \ \forall k \ \sum\nolimits_{i=1}^{c} u_{ik} = 1 \qquad (2.11.3)$$

However, the objective function J decreases as the matrix A_i tends toward being not positive definite. The usual way of obtaining a feasible solution is to constrain the determinant of A_i, allowing the matrix A_i to vary with its determinant fixed. This corresponds to optimizing the shape of the cluster while its volume remains constant:

$$\|A_i\| = \rho_i, \ \rho_i > 0, \ \forall i \qquad (2.11.4)$$

formulating the optimization problem using the Lagrange multiplier method, the following expression for A_i is obtained (Gustafson and Kessel, 1979):

$$A_i = \left[\rho_i \det(F_i) \right]^{1/n} F_i^{-1} \qquad (2.11.5)$$

where F_i is the fuzzy covariance matrix of the i th cluster given by:

$$F_i = \frac{\sum\nolimits_{k=1}^{N} (\mu_{ik})^m (x_k - v_i)(x_k - v_i)^T}{\sum\nolimits_{k=1}^{N} (\mu_{ik})^m}, \ 1 \leq i \leq c \qquad (2.11.6)$$

Given the data set X, the number of clusters c, the weighting exponent $m > 1$, the termination tolerance $\varepsilon > 0$ and volume constrains ρ_i and initializing the partition matrix U ($\left[\mu_{ik}^{(0)} \right]$) randomly, subject to (2.11.3), the Gustafson–Kessel algorithm is implemented by the following steps:

Repeat for $l = 1, 2, \ldots$

Step 1: Compute the cluster center:

$$v_i^{(l)} = \frac{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)} \right)^m x_k}{\sum_{k=1}^{N} \left(\mu_{ik}^{(l-1)} \right)^m}, \quad 1 \le i \le c. \tag{2.11.7}$$

Step 2: Compute the cluster covariance matrix F_i, according to Eq. (2.11.6)

Step 3: Compute the distances $D_{ikA_i}^2$, according to Eq. (2.11.1)

Step 4: Update the partition matrix:

$$\mu_{ik}^{(l)} = \frac{1}{\sum_{j=1}^{c} \left(D_{ikA_i} / D_{jkA_j} \right)^{2/(m-1)}} \tag{2.11.8}$$

until $\left\| U^{(l)} - U^{(l-1)} \right\| < \varepsilon$.

A more detailed description of the Gustafson–Kessel algorithm, from a mathematical point of view, can be found in the following [4, 5, 12, 16].

3. MATERIAL AND METHOD

3.1 The Dataset

The sleep EEG data used in this research were obtained from the MIT-BIH Polysomnographic Database, a collection of recordings of multiple physiologic signals during sleep. In this database, all 16 subjects were male, aged 32 to 56 (mean age 43), with weights ranging from 89 to 152 kg (mean weight 119 kg).

The database consists of 18 records. Records slp01a and slp01b are different segments of one subject's polysomnogram, separated by a gap of about one hour; records slp02a and slp02b are different segments of another subject's polysomnogram, separated by a ten-minute gap. In this study, we combine slp01a and slp01b as one record, slp01, as well as slp02a and slp02b as one record, slp02. The remaining 14 records are all from different subjects. Thus, 16 records were analyzed in this research.

The EEG electrodes placement scheme used in this database is the international 10-20 system illustrated in Figure 3.1, and Table 3.1.1 lists the EEG electrodes placement for each record. The letter and number identify the area of the scalp where the electrode is placed. The letters C and O stand for central and occipital lobes, respectively. The even numbers refer to the right hemisphere, whereas the odd number the left.

The EEG was recorded at a sampling rate of 250 Hz. The recordings are segmented

into epochs of 30 seconds and are annotated with respect to sleep stages according to the criteria of Rechtschaffen and Kales. Table 3.1.2 summarizes the stage distribution of each record, in which, 'Epochs', 'W', 'N1', 'N2', 'N3', 'R' and 'M' stand for 'the number of total epochs of 'wake', 'NREM stage 1', 'NREM stage 2', 'NREM stage 3', 'REM' and 'movement time' respectively.

Figure 3.1: 10-20 system. This figure is taken from http://en.wikipedia.org/wiki/File:21_electrodes_of_International_10-20_system_for_EEG.svg
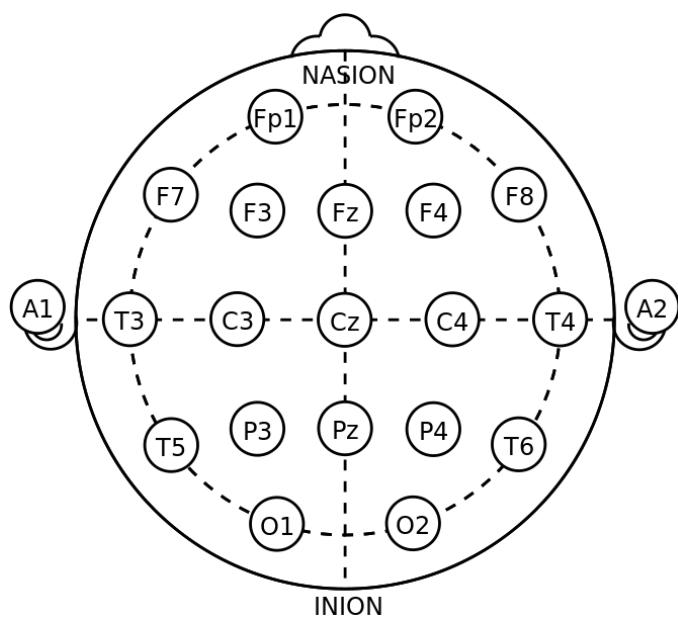

Table 3.1.1: EEG electrodes placement of each record

Record	Electrode-Pair
slp01	C4-A1

slp02	O2-A1
slp03	C3-O1
slp04	C3-O1
slp14	C3-O1
slp16	C3-O1
slp32	C4-A1
slp37	C4-A1
slp41	C4-A1
slp45	C3-O1
slp48	C3-O1
slp59	C3-O1
slp60	C3-O1
slp61	C3-O1
slp66	C3-O1
slp67x	C3-O1

Table 3.1.2: The stage distribution of each record

Record	Epochs	W	N1	N2	N3	R	M
slp01	600	187	28	233	113	38	1
slp02	630	151	32	325	7	106	9
slp03	720	151	105	312	78	74	0
slp04	720	162	60	442	33	23	0
slp14	714	321	188	126	42	36	1
slp16	694	316	108	181	24	65	0
slp32	640	394	27	159	60	0	0
slp37	698	75	21	591	0	11	0
slp41	780	229	230	218	13	90	0
slp45	760	119	54	399	103	81	4
slp48	760	213	241	272	2	31	1
slp59	458	140	105	98	80	35	0
slp60	710	286	344	49	0	31	0
slp61	720	124	88	326	103	79	0
slp66	439	175	143	116	5	0	0
slp67x	154	72	41	40	1	0	0

3.2   Method

In this study, we developed a method of wake/sleep scoring for particular subjects, as is illustrated in the block diagram presented in Figure 3.2. The method is unsupervised and consists of 5 consecutive steps: segmentation, feature extraction, feature reduction, clustering and wake/sleep scoring. Step 1: Divide the EEG into non-overlapping 10s segments (3 segments in each 30s epoch) rather than the traditional 30s epochs. In this way, by score attached to a given epoch is determined by the majority score of its segments, it will improve the scoring accuracy in the last step (wake/sleep scoring). Also, by increasing the number of segments (by 3) in the dataset, the data generated Step 3 (feature reduction) will be easier for clustering in Step 4 (clustering). Step 2: For each segment, 19 features are extracted, and a feature matrix of the entire sleep record is formed. Step 3: The feature matrix is scaled and PCA is applied to project the data onto a lower dimension space (called principal components space) that is easier for identifying patterns in the data while retaining most of its information. The objective of classification in Step 4 is to separate wake and sleep. Step 4: A fuzzy clustering algorithm, the Gustafson-Kessel Algorithm is applied to the data set (in the principal components space) determined in Step 3. Step 5: Wake/sleep scoring is obtained by retrieving one characteristic feature of wake stage as a wake stage indicator. The last 4 steps will be described in detail in the following chapters.

Figure 3.2: Schematic overview of our method for wake/sleep scoring
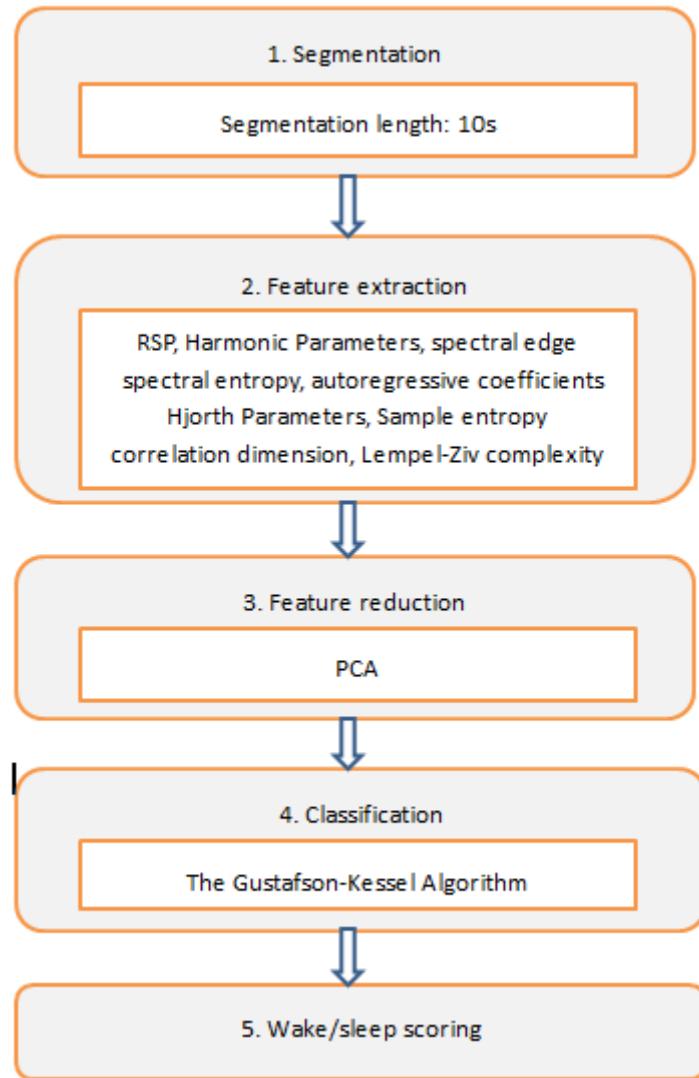
3.3   Feature Extraction

In the context of EEG analysis, a feature is an index that mathematically characterizes one aspect of the EEG signal. In feature extraction step, 19 features are extracted in each 10s segment as described in the following.

According to AASM Manual, each sleep stage is essentially visually distinguished by some spectral property. In this study, the power spectral density (PSD) is estimated using Welch's method (2.1.1) in the frequency range between 0.5Hz and 45Hz. Spectral features are obtained by dividing the spectrum into 7 frequency sub-bands as presented in Table 3.3.1. For each sub-band, the relative spectral power (RSP) given by the ratio between the sub-band spectral power and the total power is computed. Harmonic parameters (2.1.2) that consist of center frequency, bandwidth and magnitude of the PSD at the center frequency, 90% spectral edge (2.1.3) (the frequency under which 90% of the spectrum is contained) and spectral entropy (2.1.4) are also extracted in the frequency domain. The 1st autoregressive coefficient (2.1.5) turns out to be a reliable feature for distinguishing wake and sleep. An AR model of order 5 is chosen and by using Burg's method the parameters of the AR model are estimated. Hjorth parameters (2.1.6) that consist of activity, mobility and complexity are calculated from the time series. Nonlinear time series analyses, such as sample entropy (2.1.7, with pattern length m =2, tolerance r =0.2SD), correlation dimension (2.1.8, with embedding dimension m =20, time lag τ =10) and Lempel-Ziv complexity (2.1.9) are also performed.

With the x-axis denoting the records and the y-axis denoting the value of a particular feature, the performance of the 19 features are shown in Figure 3.3.1 to 3.3.19 (the difference of wake and sleep is significant if p<0.05; if not, otherwise). In these figures, the upper bound, median and lower bound of each error bar corresponds to

the 75 th percentile, 50 th percentile and 25 th percentile, respectively of the feature

either in wake or sleep (X percentile means that X percent of the data is located within

this range).

Table 3.3: Spectral sub-bands used in RSP computation

sub-bands	range (Hz)
Delta 1	0.5 - 2.5
Delta 2	2.5 - 4
Theta 1	4 - 6
Theta 2	6 - 8
Alpha	8 - 12
Beta 1	12 - 20
Beta 2	20 - 45

Figure 3.3.1: Comparison of center frequency in wake and sleep (p<0.05)
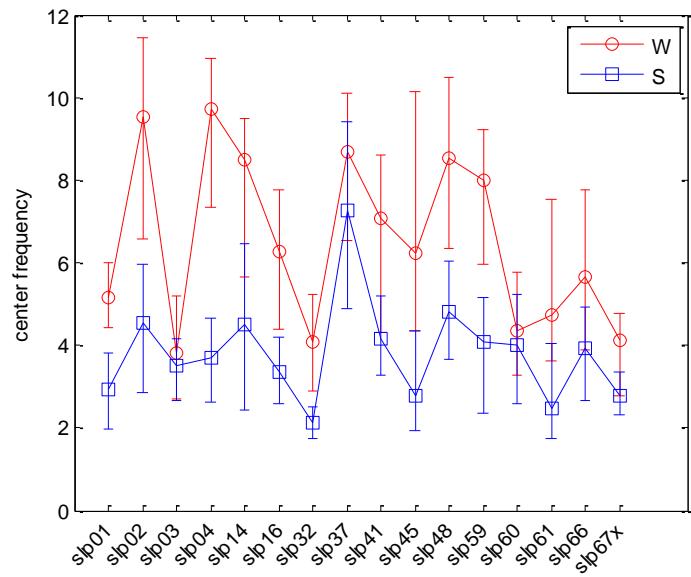
Figure 3.3.2: Comparison of bandwidth in wake and sleep (p<0.05; except slp14 (p=0.1227) and slp60 (p=0.0729))
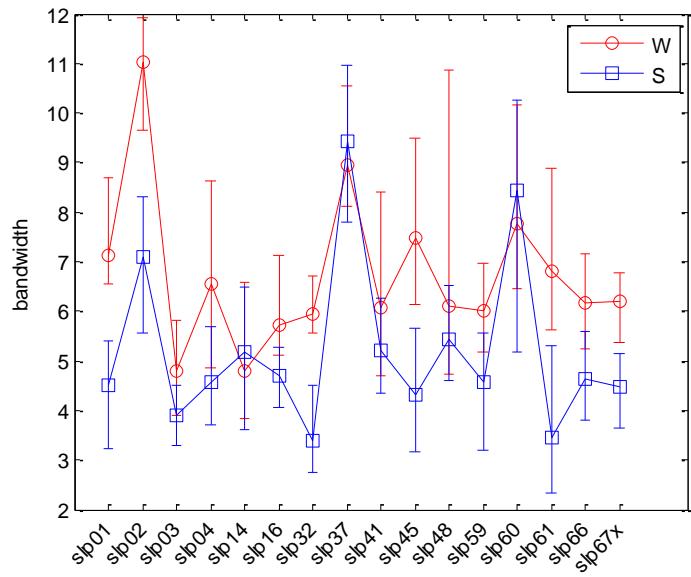
Figure 3.3.3: Comparison of the PSD at the center frequency in wake and sleep (p<0.05; except slp02 (p=0.1904), slp16 (p=0.0569) and slp59 (p=0.5139))
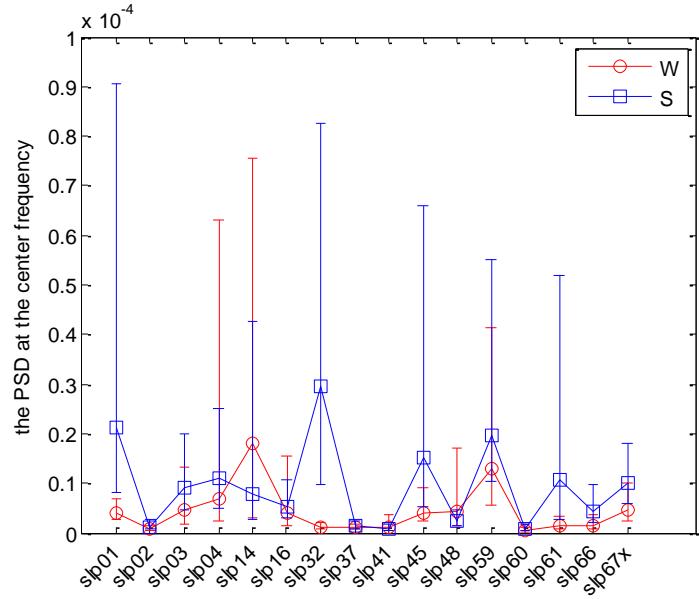
Figure 3.3.4: Comparison of RSP Delta 1 in wake and sleep (p<0.05; except slp03 (p=0.2010) and slp67x (p=0.4176))
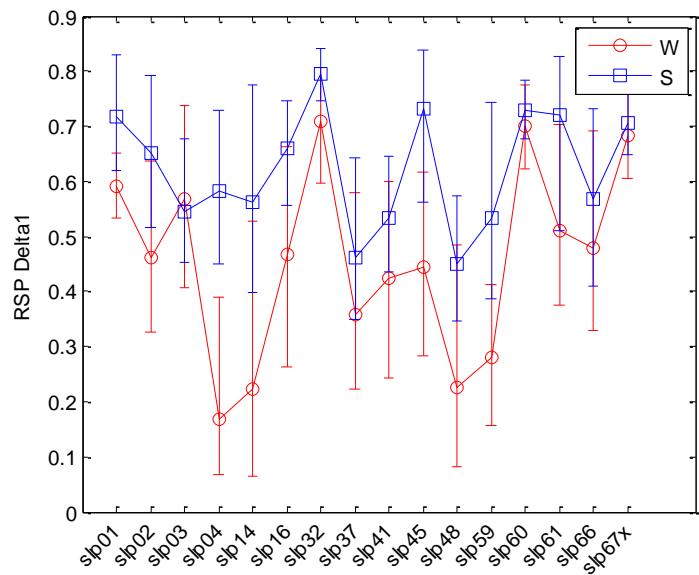
Figure 3.3.5: Comparison of RSP Delta2 in wake and sleep (p<0.05)
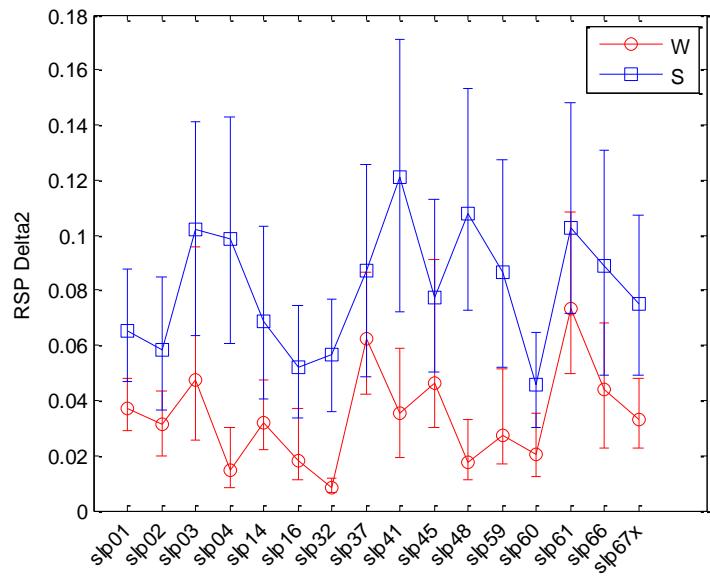
Figure 3.3.6: Comparison of RSP Theta1 in wake and sleep (p<0.05; except slp37 (p=0.1862) and slp61 (p=0.4176))
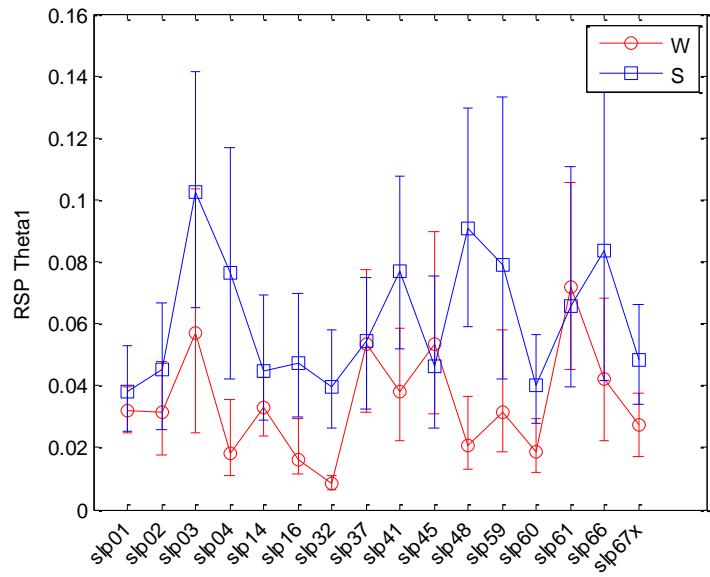
Figure 3.3.7: Comparison of RSP Theta2 in wake and sleep (p<0.05; except slp01 (p=0.1070) and slp59 (p=0.8035))
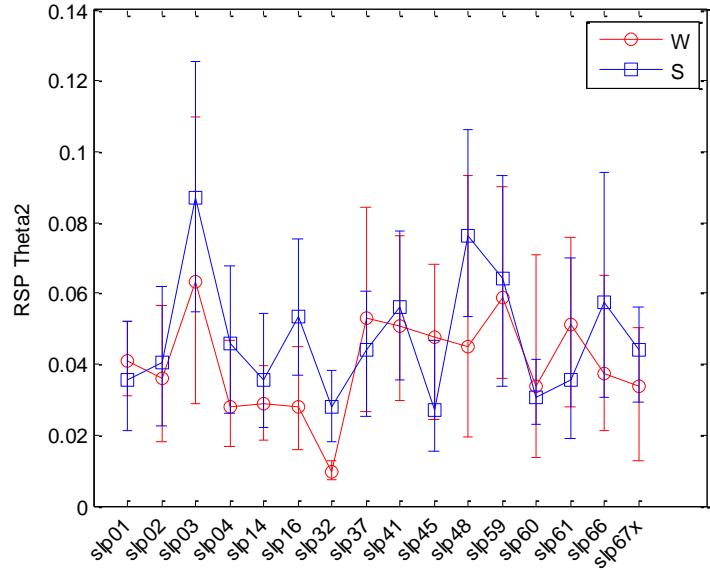
Figure 3.3.8: Comparison of RSP Alpha in wake and sleep (p<0.05)
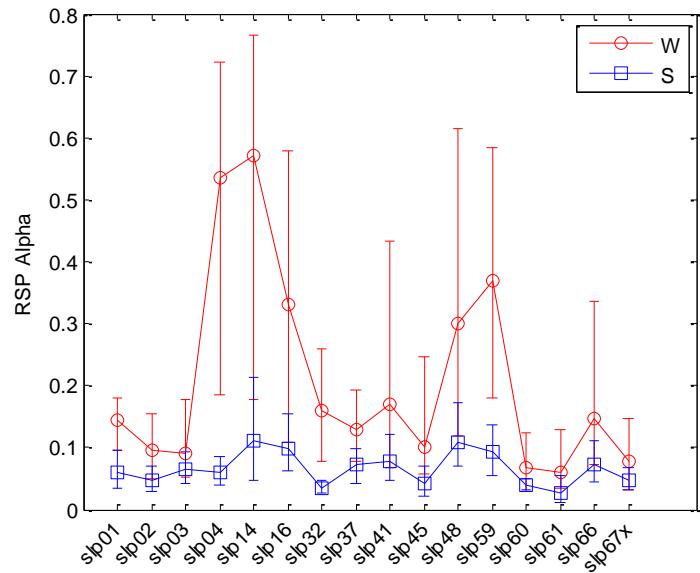
Figure 3.3.9: Comparison of RSP Beta1 in wake and sleep (p<0.05)
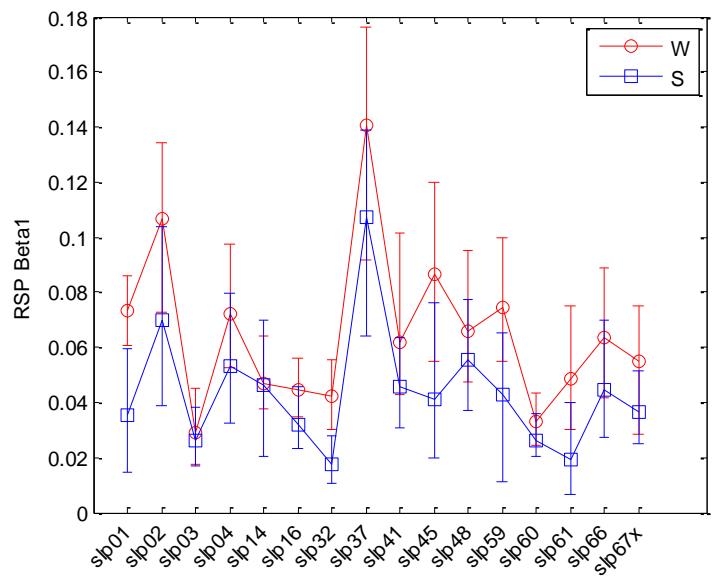
Figure 3.3.10: Comparison of RSP Beta2 in wake and sleep (p<0.05)
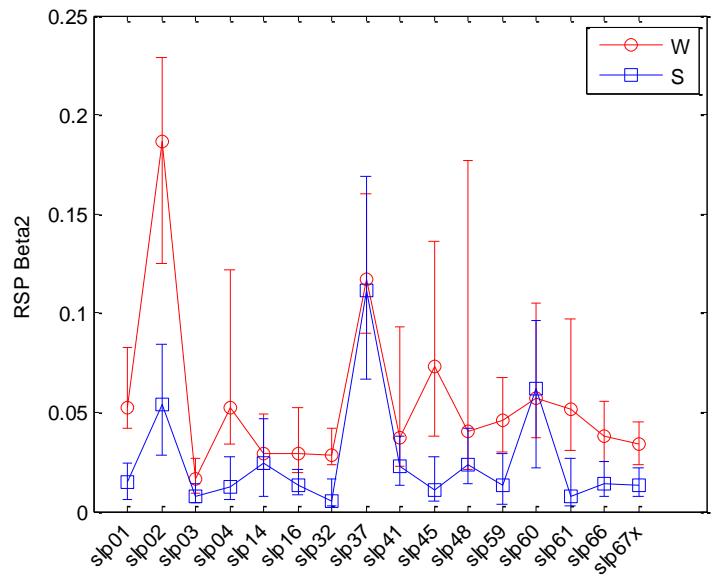
Figure 3.3.11: Comparison of activity in wake and sleep (p<0.05, except slp16 (p=0.1359))
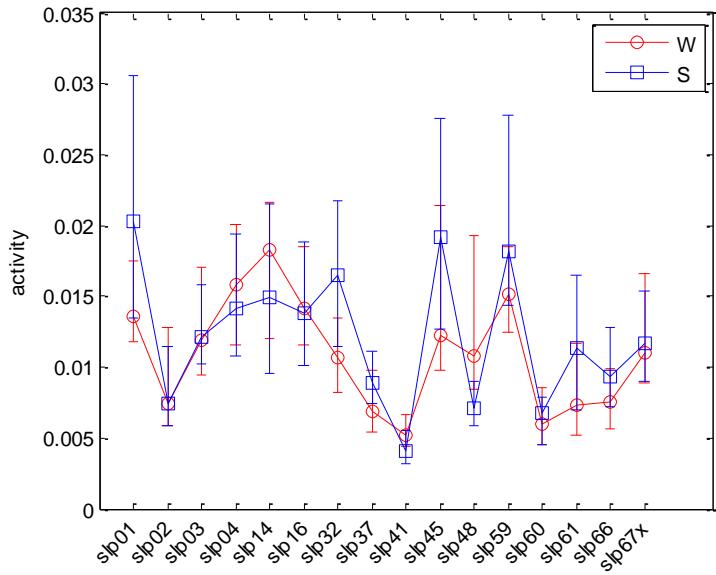
Figure 3.3.12: Comparison of mobility in wake and sleep (p<0.05; except slp37 (p=0.1849) and slp60 (p=0.1774))
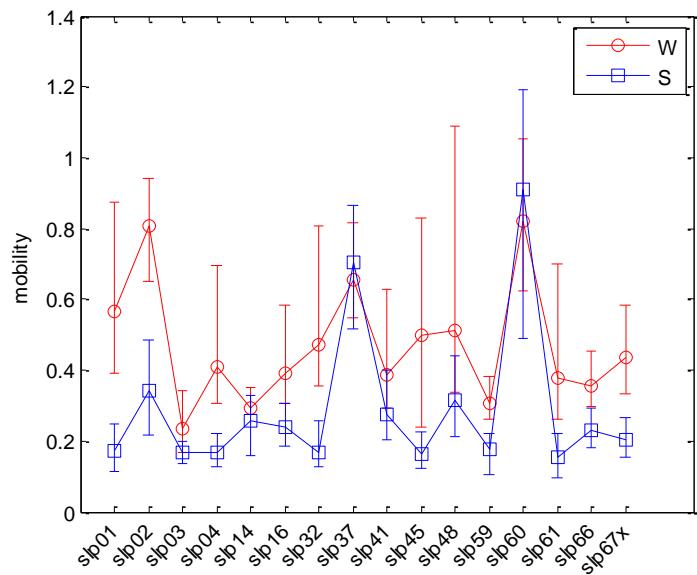
Figure 3.3.13: Comparison of complexity in wake and sleep (p<0.05)
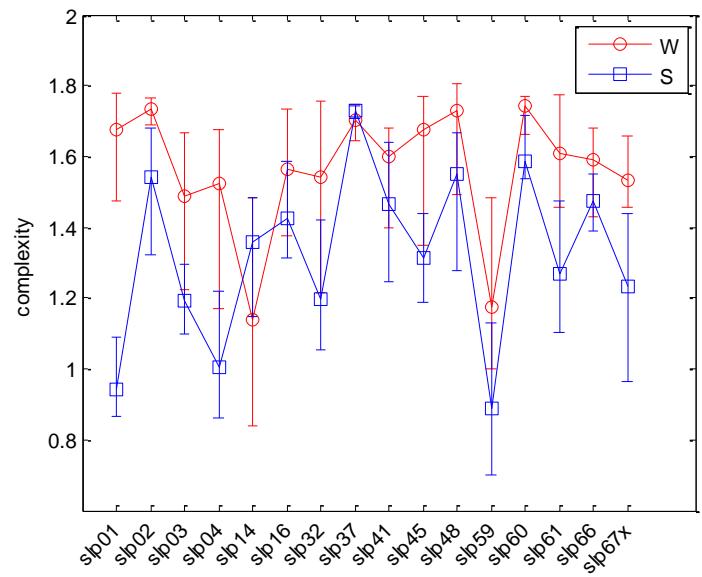
Figure 3.3.14: Comparison of sample entropy in wake and sleep (p<0.05)
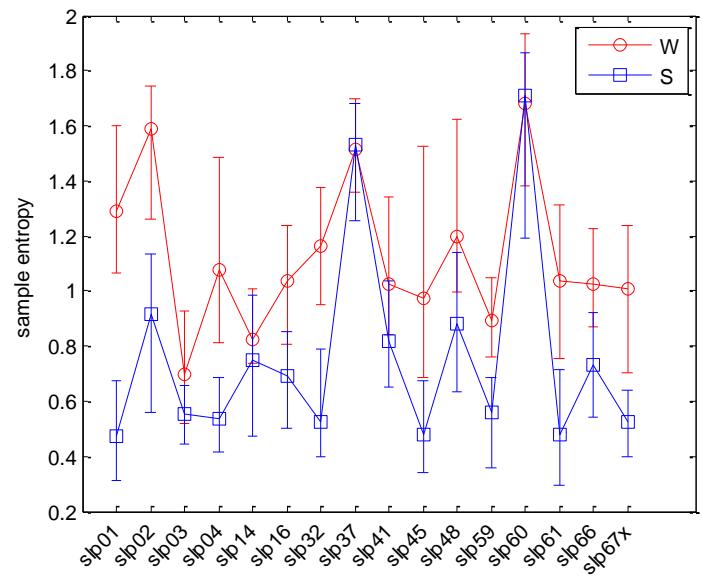
Figure 3.3.15: Comparison of the 1st autoregressive coefficient in wake and sleep (p<0.05)
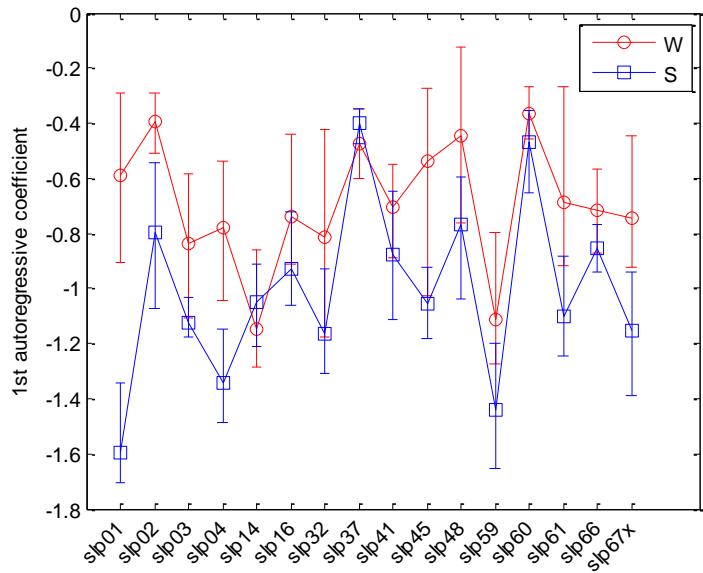
Figure 3.3.16: Comparison of correlation dimension in wake and sleep (p<0.05)
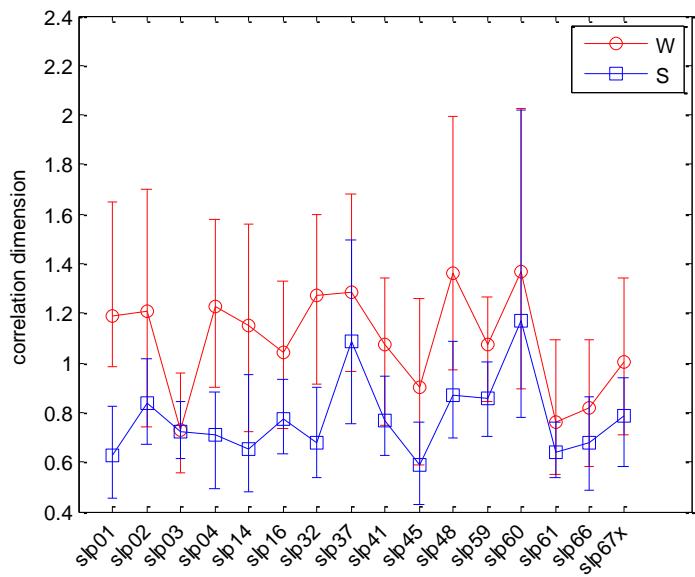
Figure 3.3.17: Comparison of Lempel-Ziv complexity in wake and sleep (p<0.05)
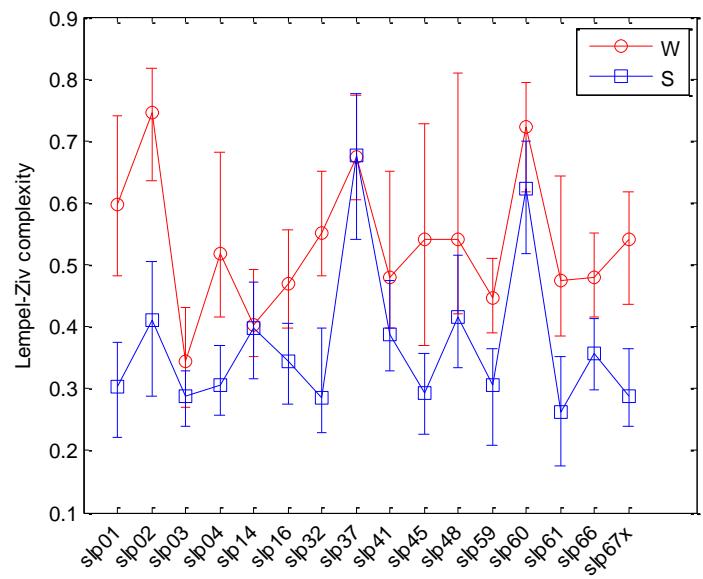
Figure 3.3.18: Comparison of 90% spectral edge in wake and sleep (p<0.05)
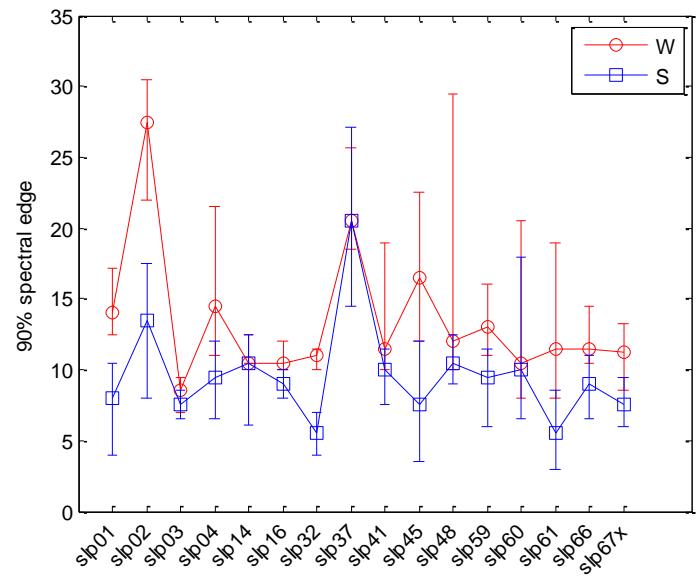
Figure 3.3.19: Comparison of spectral entropy in wake and sleep (p<0.05; except slp41 (p=0.5436), slp60 (p=0.5429) and slp67x (p=0.2620))
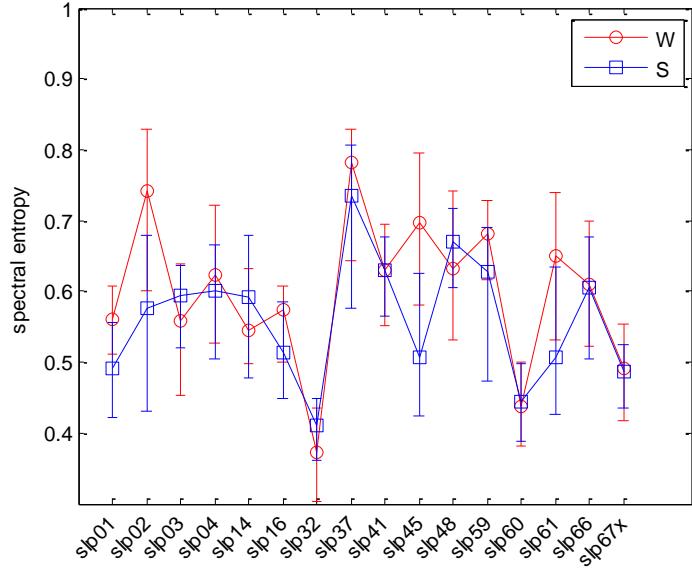
3.4 Feature Reduction

Next, PCA (2.10) is applied in order to extract the relevant information for separating wake and sleep while reducing the redundancy in the feature set. Also, by reducing the number of features while retaining most of the information, it helps to reduce the "curse of dimensionality", where the sparsity of data in a high dimensional feature space would make traditional statistical methods fail.

In the previous section, feature extraction provides a feature vector to each 10s segment, where each feature vector consists of 19 features. A feature matrix X is

formed for each record: $X \in \mathbb{R}^{n \times N}$, where n is the number of features and N is the number of 10s segments in a record. The column i of X represents the feature vector of the i th segment. As features are measured in different scales, the feature vector needs to be normalized, feature-by-feature before applying PCA. Otherwise, features with larger scales would dominate in the PCA. Thus, each row of X (a feature) is normalized by:

$$\hat{X}_j = \frac{X_j - X_{j,\min}}{X_{j,\max} - X_{j,\min}}, \ j = 1, 2, ..., n \tag{3.4.1}$$

where $X_{j,\min}$ and $X_{j,\max}$ denote the minimum and maximum of the j th row respectively.

Then, PCA is applied to transform the normalized feature matrix $\hat{X}$ to S by:

$$\hat{S} = U^T \cdot \hat{X} \tag{3.4.2}$$

where $\hat{S} \in \mathbb{R}^{n \times N}$. The i th row of $\hat{S}$ is the projection of $\hat{X}$ on the i th principal component sorted in order of significance. The significance of a principal component is measured by its singular value and the number of principal components that we select is determined by the cumulative contribution rate of the m th principal component, given by:

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i}, \ m \leq n \tag{3.4.3}$$

where λ_i is the i th eigenvalue.

33

Take record slp01 as an example, Table 3.4.1 shows its accumulative contribution rate

of each principal component. From Table 3.4.1, we know the first 6 principal

components contain most of the information (95.59%) of the data. Figure 3.4.1

illustrates the data from slp01 projected onto the first 3 principal components.

In this study, we take the first p principal components that contain 95% of the

information. This is done simply by taking the first p rows of $\hat{S}$, denoted by S

($S \in \mathbb{R}^{p \times N}$). Thus, S serves as the refined feature matrix for classification in the

upcoming step.

Table 3.4.1: Cumulative contribution rate of each principal component of slp01

i [1]	r [2]
1	0.7091
2	0.8455
3	0.8906
4	0.9221
5	0.9416
6	0.9559
7	0.9679
8	0.9764
9	0.9832
10	0.9879
11	0.9911
12	0.9938
13	0.9959
14	0.9975
15	0.9987
16	0.9996
17	1.0000
18	1.0000

19	1.0000

[1]: i stands for the ith principal component.

[2]: r stands for cumulative contribution rate.

Figure 3.4.1: The data from slp01 projected onto the first 3 principal components
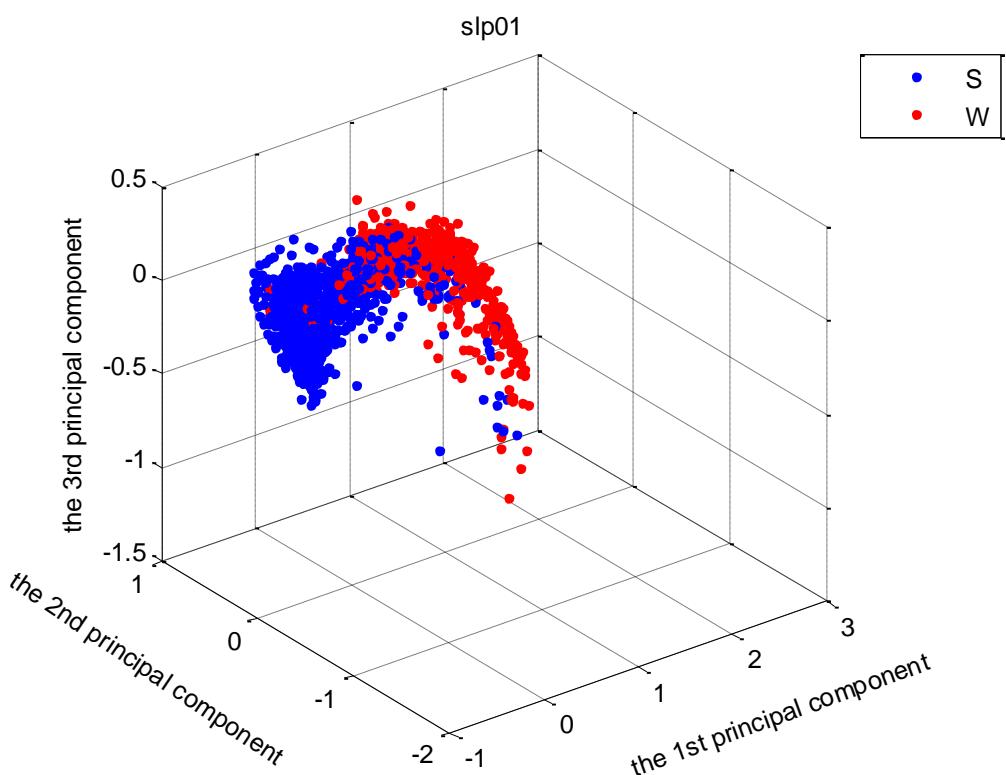
3.5 Clustering Analysis

In Figure 3.4.1, the data structure that PCA generates has two clusters, one belongs to wake, and the other belongs to sleep. This motivates us to separate wake and sleep

using a method of clustering analysis.

The objective of clustering analysis is to organize data into groups according to similarities (often measured by means of a distance norm) among them. Clustering analysis is an unsupervised method, that is, it doesn't use class identifiers a priori. There are two main categories of clustering methods; one is hard clustering and the other is fuzzy clustering. In hard clustering, data is divided into distinct clusters, where each data element belongs to one and only one cluster. Hence the clusters in a hard clustering approach are disjoint. In fuzzy clustering, on the other hand, data elements can belong to several clusters simultaneously, with different degrees of membership which indicate the strength of the association between that data element and a particular cluster.

As shown in Figure 3.4.1, fuzzy clustering is more natural than hard clustering in the situation of wake and sleep separation, as objects on the boundaries between these two classes are not disjoint, however conjoined.

Among fuzzy clustering algorithms, GK-FCM (2.11) is selected. It extends fuzzy c-means (FCM) by employing an adaptive distance norm, in order to detect clusters with different geometrical shapes. In this study, GK-FCM is implemented by using the Fuzzy Clustering Toolbox in Matlab.

The dataset to be clustered is the refined feature matrix S ($S \in \mathbb{R}^{p \times N}$) generated by PCA in the previous section. Each segment can be viewed as a data point of a p

dimensional space. The input parameters of GK-FCM that we set are: the number of clusters $c = 2$, the weighting exponent $m = 2$, the termination tolerance $\varepsilon = 10^{-6}$ and the volume constrains $\rho_i = 1$. the output of the method is a membership matrix U ($U \in \mathbb{R}^{2 \times N}$). An element u_{ij} ($u_{ij} \in [0,1]$) of this matrix represents the grade of membership of the j th segment in cluster c_i. The larger membership values indicate higher confidence in the assignment of the data to the cluster. Then, we assign the j th segment the number $\hat{i} = \arg\max_i \mu_{ij}$. In this way, the dataset S is clustered into two clusters labeled 1 and 2. Figure 3.5.1 illustrates the clustering result of slp01.

Figure 3.5.1: Clustering result of slp01
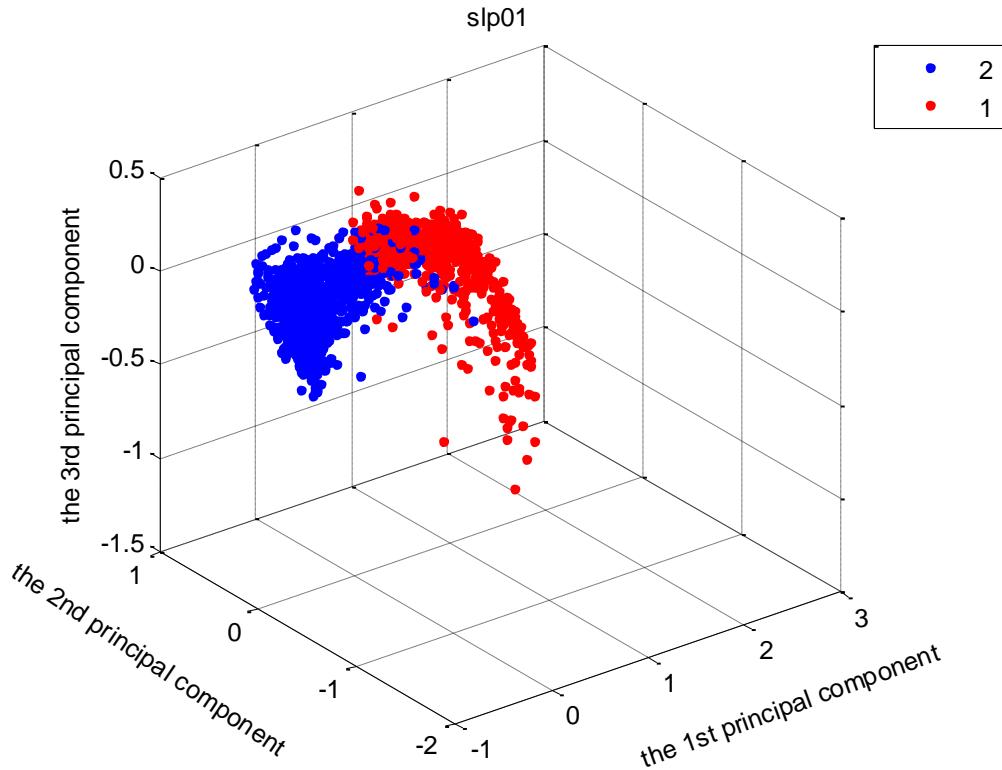
3.6 Wake/Sleep Scoring

Wake/sleep scoring is obtained by labeling the two clusters of dataset S either "wake" or "sleep". In this study, this is done by retrieving one of the characteristic features of the wake stage: Relative Spectral Power of Alpha (RSP Alpha). According to ASSM Manual, wake stage is characterized by alpha rhythm in EEG. By computing the mean of RSP Alpha in each cluster, the cluster with the higher mean RSP Alpha is labeled as "Wake", the other as "Sleep". After all the segments are labeled, we label the 30s epochs according to the majority of the 3 sub-epochs label.

4   RESULT

After all the epochs are labeled as "wake" or "sleep", the scoring results in each record are compared with actual stages that are labeled by an expert. The confusion matrix of each record is obtained, and the mean of RSP Alpha in each cluster (labeled 1 or 2) is calculated. Table 1~16 summarize our classification results for the 16 records. The last row of these tables contains the mean of RSP Alpha in each cluster. Recall our scoring rule in 3.6: the clusters with larger RSP Alpha are labeled as "wake", the other as "sleep".

Sensitivity, specificity and accuracy are three metrics that are used to measure the performance of a classifier. They are defined as:

$$sensitivity = \frac{TP}{TP + FN} \qquad\qquad (4.1)$$

$$specificity = \frac{FP}{FP + TN} \qquad\qquad (4.2)$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad\qquad (4.3)$$

where TP, FN, FP and TN stand for true positive, false negative, false positive and true negative respectively. Table 4.17 summaries the sensitivity, specificity and accuracy of each record. The accuracies of 14 records are satisfactory (above 75%), slp37 (36.10%) and slp60 (57.46%) are not as satisfactory.

Table 4.1: Classification results for slp01

		clusters	
		1	2
Actual	Wake	**168**	19
Stage	Sleep	25	**388**
mean of RSP Alpha		0.1367	0.0700

Table 4.2: Classification results for slp02

		clusters	
		1	2
Actual	Wake	**131**	20
Stage	Sleep	91	**388**
mean of RSP Alpha		0.1046	0.0441

Table 4.3: Classification results for slp03

		clusters	
		1	2
Actual	Wake	**92**	59
Stage	Sleep	55	**514**
mean of RSP Alpha		0.1899	0.0674

Table 4.4: Classification results for slp04

		clusters	
		1	2
Actual	Wake	20	**142**
Stage	Sleep	**504**	54
mean of RSP Alpha		0.0598	0.4801

Table 4.5: Classification results for slp14

		clusters	
		1	2
Actual	Wake	80	**241**
Stage	Sleep	**302**	91
mean of RSP Alpha		0.0945	0.5682

Table 4.6: Classification results for slp16

		clusters	
		1	2
Actual	Wake	**231**	85
Stage	Sleep	76	**302**
mean of RSP Alpha		0.3979	0.0859

Table 4.7: Classification results for slp32

		clusters	
		1	2
Actual	Wake	53	**341**
Stage	Sleep	**232**	14
mean of RSP Alpha		0.0415	0.1947

Table 4.8: Classification results for slp37

		clusters	
		1	2
Actual	Wake	21	**54**
Stage	Sleep	**198**	425
mean of RSP Alpha		0.0332	0.1024

Table 4.9: Classification results for slp41

		clusters	
		1	2
Actual	Wake	**194**	35
Stage	Sleep	91	**460**
mean of RSP Alpha		0.2582	0.0956

Table 4.10: Classification results for slp45

		clusters	
		1	2
Actual	Wake	38	**81**
Stage	Sleep	**600**	41
mean of RSP Alpha		0.0513	0.1502

Table 4.11: Classification results for slp48

		clusters	
		1	2
Actual	Wake	46	**167**
Stage	Sleep	**491**	56
mean of RSP Alpha		0.1206	0.3927

Table 4.12: Classification results for slp59

		clusters	
		1	2
Actual	Wake	38	**102**
Stage	Sleep	**288**	30
mean of RSP Alpha		0.1001	0.4673

Table 4.13: Classification results for slp60

		clusters	
		1	2
Actual	Wake	**183**	103
Stage	Sleep	199	**225**
mean of RSP Alpha		0.0833	0.0344

Table 4.14: Classification results for slp61

		clusters	
		1	2
Actual	Wake	52	**72**
Stage	Sleep	**554**	42
mean of RSP Alpha		0.0342	0.1310

Table 4.15: Classification results for slp66

		clusters	
		1	2
Actual	Wake	69	**106**
Stage	Sleep	**230**	34
mean of RSP Alpha		0.0680	0.3369

Table 4.16: Classification results for slp67x

		clusters	
		1	2
Actual	Wake	30	**42**
Stage	Sleep	**75**	7
mean of RSP Alpha		0.0596	0.0942

Table 4.17: Sensitivity, specificity and accuracy of each record

| Record	sensitivity	specificity	accuracy
slp01	89.84%	93.95%	92.67%
slp02	86.75%	81.00%	82.38%
slp03	60.93%	90.33%	84.17%
slp04	87.65%	90.32%	89.72%
slp14	75.08%	76.84%	76.05%
slp16	73.10%	79.89%	76.80%
slp32	86.55%	94.31%	89.53%
slp37	72.00%	31.78%	36.10%
slp41	84.72%	83.48%	83.85%
slp45	68.07%	93.60%	89.61%
slp48	78.40%	89.76%	86.58%
slp59	72.86%	90.57%	85.15%
slp60	63.99%	53.07%	57.46%
slp61	58.06%	92.95%	86.94%
slp66	60.57%	87.12%	76.54%
slp67x	58.33%	91.46%	75.97%
average	73.56%	82.53%	79.35%

5   CONCLUSION AND FUTURE WORK

From the previous section, the results for most subjects (14 out of 16) in the database are satisfactory, although the number of epochs in wake and sleep in some cases are highly unbalanced. We conclude that, at least in this database, the method we propose for unsupervised wake/sleep scoring is applicable to most individuals. The future work, on one hand, is to verify this conclusion by testing more datasets; including selecting more relevant features and by applying more feasible techniques (in PSD estimation, dimension reduction and unsupervised learning), to improve the results.

This thesis shows that by extracting a collection of features and then by applying dimension reduction techniques, it is possible to generate a data structure in which different sleep stages belongs to fairly distinctive clusters. Such a data structure is appropriate for applying clustering analysis. According to AASM Manual, sleep is subdivided into NREM (including N1, N2 and N3), REM and movement time. After separating wake and sleep, the question facing us is whether or to what extent it is possible to further separate sleep into NREM and REM based on a similar idea. To this end, because NREM N1 is similar to REM in EEG, it may be necessary to use other biosignals such as EOG and EMG to facilitate classification. Based on our research, EMG features, such as center frequency, bandwidth, sample entropy, can provide useful information for separating NREM and REM due to the fact that REM presents low chin EMG tone.

45

To achieve complete sleep staging with an unsupervised approach, a hierarchical classification methodology is illustrated in Figure 5.1 maybe the most appropriate.

Figure 5.1: Hierarchical sleep stage classification
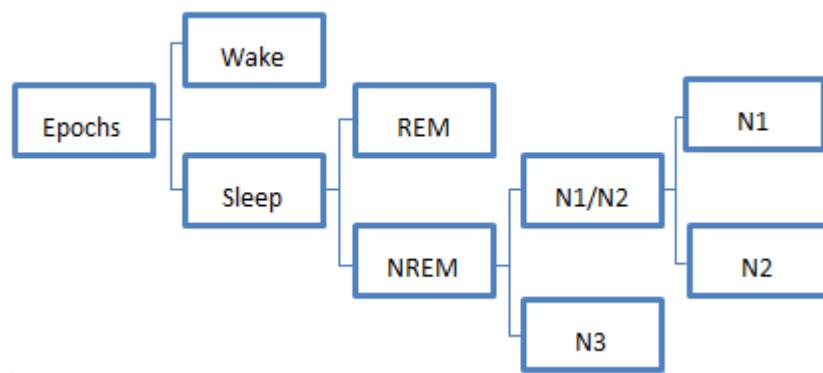
BIBLIOGRAPHY

[1] Abraham Lempel, Jacob Ziv. 1976. "On the Complexity of Finite Sequences." *IEEE TRANSACTIONS ON INFORMATION THEORY* 22(1): 75-81.

[2] Alcaraz, Raúl, Abásolo, Daniel E., Hornero, Roberto and Rieta, José Joaquín. 2010. "Optimal parameters study for sample entropy-based atrial fibrillation organization analysis." *Computer Methods and Programs in Biomedicine* 99(1): 124-132.

[3] Anurak Thungtong. 2008. "REAL-TIME AUTOMATED SLEEP SCORING OF NEONATES." MS Thesis, Case Western Reserve University.

[4] Babuka, R.; Van der Veen, P. J.; and Kaymak, U. 2002. "Improved covariance estimation for Gustafson-Kessel clustering." *In Proceedings of 2002 IEEE International Conference on Fuzzy Systems* 2 (5): 1081-1085. Honolulu, Hawaii.

[5] Balazs Balasko, Janos Abonyi, Balazs Feil. "Fuzzy Clustering and Data Analysis Toolbox   For   Use   with   Matlab."   Online.   Available:

http://www.sunfinedata.com/wp-content/uploads/2009/10/FuzzyClusteringToolbox.pd

f

[6]   Clustering   and   Data   Analysis   Toolbox.   Online.   Available:

http://www.mathworks.com/matlabcentral/fileexchange/7473-clustering-and-data-anal

ysis-toolbox

[7] Dong, J.; Liu, D.; Zhang, C.; Ma, J.; Wang, G.; Guo, D.; Liu, Y.; Zhong, H.; Zhang,

J.; Peng, C.-K. and Fang, J. 2010. "Automated Sleep Staging Technique Based on the

Empirical Mode Decomposition Algorithm: a Preliminary Study." *Advances in*

Adaptive Data Analysis 2 (2): 267-276.

[8] Ebrahimi, F.; Mikaeili, Mohammad; Estrada, E.; Nazeran, H. 2008. "Automatic

sleep stage classification based on EEG signals by using neural networks and wavelet

packet coefficients." *Engineering in Medicine and Biology Society, 2008. EMBS 2008.*

30th Annual International Conference of the IEEE: 1151-1154.

[9] Edson Estrada, Homer Nazeran, Khosrow Behbehani, John Burk, and Edgar Lucas.

2008. "Correlation dimension analysis of simultaneous eeg and ecg signals for

automated sleep stage classification." *BIOMEDICAL ENGINEERING Recent Developments.*

[10] Figueroa Helland VC, Gapelyuk A, Suhrbier A, Riedl M, Penzel T, Kurths J, and Wessel N. 2010. "Investigation of an automatic sleep stage classification by means of multiscorer hypnogram." *Methods Inf Med.* 49(5): 467-72.

[11] Gath, I.; Geva, A.B. 1989. "Unsupervised optimal fuzzy clustering." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (7): 773-780.

[12] Graves, D.; Pedrycz, W. 2007. "Fuzzy C-Means, Gustafson--Kessel FCM, and Kernel-based FCM: a comparative study." *Anal. and Des. of Intel. Sys. using SC Tech., ASC* 41: 140–149.

[13] Günes, S.; Polat, K.; and Yosunkaya, S. 2010. "Efficient sleep stage recognition system based on EEG signal using k-means clustering based feature weighting." *Expert Syst. Appl.* 37 (12): 7922-7928.

[14] Huang, J.-R.; Fan, S.-Z.; Abbod, M. F.; Jen, K.-K.; Wu, J.-F. and Shieh, J.-S. 2013. "Application of Multivariate Empirical Mode Decomposition and Sample Entropy in EEG Signals via Artificial Neural Networks for Interpreting Depth of Anesthesia." *Entropy* 15(9): 3325-3339.

[15] Hugo Simões, Gabriel Pires, Urbano Nunes, and Vitor Silva. "FEATURE EXTRACTION AND SELECTION FOR AUTOMATIC SLEEP STAGING USING EEG." Online. Available: http://www2.isr.uc.pt/~gpires/papers/ICINCO2010.pdf?

[16] Hui WANG, Chengjun HUANG, Linpeng YAO, Yong QIAN, and Xiuchen JIANG. 2011. "Application of Gustafson-Kessel clustering algorithm in the pattern recognition for GIS." *PRZEGLĄD ELEKTROTECHNICZNY (Electrical Review)* 87(1): 215-219.

[17] Iber C, Ancoli-Israel S, Chesson A, and Quan S. 2007. *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications.* Westchester, IL: American Academy of Sleep Medicine.

[18] J. Fell, J. Röschke, K. Mann, and C. Schäffner. 1996. "Discrimination of sleep

stages: a comparison between spectral and nonlinear EEG measures."
Electroencephalogr Clin Neurophysiol 98 (5): 401–410.

[19] J.S Richman, J.R. Moorman. 2000. "Physiological time-series analysis using approximate entropy and sample entropy." *Am. J. Physiol. Heart Circ. Physiol.* 278(6): 2039-2049.

[20] Kavita Mahajan, M R Vargantwar, and Sangita M Rajput. 2012. "Classification of EEG using PCA, ICA and Neural Network." *IJCA Proceedings on International Conference in Computational Intelligence (ICCIA2012)* 6: 80 – 83.

[21] Lindsay I Smith. 2002. "A tutorial on principal components analysis." Online. Available: www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf

[22] Mateo Aboy, Roberto Hornero, Daniel Abásolo, and Daniel Álvarez. 2006. "Interpretation of the Lempel-Ziv Complexity Measure in the Context of Biomedical Signal Analysis." *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING* 53(11): 2282-2288.

[23] MIT-BIH Polysomnographic Database. Online. Available: http://www.physionet.org/physiobank/database/slpdb/

[24] Saeed V. Vaseghi. 2000. *Advanced Digital Signal Processing and Noise Reduction, Second Edition.* Chichester, UK: John Wiley & Sons Ltd.

[25] Saeid Sanei, J.A. Chambers. 2007. *EEG SIGNAL PROCESSING.* Chichester, UK: John Wiley & Sons Ltd.

[26] Sergios Theodoridis, Konstantinos Koutroumbas. 2009. *Pattern Recognition, Fourth Edition.* London: Academic Press.

[27] Shing-Tai Pan, Chih-En Kuo, Jian-Hong Zeng, Sheng-Fu Liang. 2012. "A transition-constrained discrete hidden Markov model for automatic sleep staging." *Biomed Eng Online.* Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3462123/

[28] The WFDB Toolbox for Matlab. Online. Available: http://physionet.org/physiotools/matlab/wfdb-app-matlab/

[29] Todd Levy, Kenneth A. Loparo, and Mark Scher. "Automated Scoring of Neonatal Sleep." Online. Available: http://systemsbiology.case.edu/publications/sleep/ieee_neosleep_submit_05.pdf

[30] Van Hese, P.; Philips, W.; De Koninck, J.; Van De Walle, R.; and Lemahieu, I. 2001. "Automatic detection of sleep stages using the EEG." *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE* 2: 1944-1947.

[31] Zhou Peng, Li Xiangxin, Zhang Yi, and Xue Ranting. 2012. "Analysis of sleep staging based on Individual difference with Principal Component Analysis and Support Vector Machine." Online. Available: http://www.paper.edu.cn/releasepaper/content/201201-785