8-2014

Improving Golf Putt Performance with Statistical Learning of EEG Signals

Qing Guo
University of Arkansas, Fayetteville

Improving Golf Putt Performance with Statistical Learning of EEG Signals

Improving Golf Putt Performance with Statistical Learning of EEG Signals

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Electrical Engineering

By

Qing Guo
Shenzhen University
Bachelor of Science in Electronic Science and Technology, 2011

August 2014
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

Dr. Jingxian Wu
Thesis Director

Dr. Baohua Li
Committee Member

Dr. Shengfan Zhang
Committee Member

ABSTRACT

In this thesis, a machine learning based method is proposed to predict the putt outcomes of golfers based on their electroencephalogram (EEG) data. The method can be used as a core building block of a brain-computer interface, which is designed to provide guidance to golf players based on their EEG patterns. The proposed method includes three steps. First, multi-channel 1-second EEG trials were extracted during golfers' preparation of putting. Second, different features are calculated such as correlation coefficient, power spectrum density and coherence, which are used as features for the classification algorithm. To predict golfers' performance, the support vector machine algorithm is used to classify the EEG patterns into two categories corresponding to successful and non-successful putts. The proposed approach utilizes a large number of features extracted from the EEG signals, and it is capable of providing adequate prediction that could help golfers to improve their performances.

ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor Dr. Jingxian Wu for the support of my master study and research, for his caring, patience, and persistent help. I could not have imagined having a better advisor and mentor for my master study.

Second, I am also extremely grateful to my committee members, Dr. Baohua Li and Dr. Shengfan. Zhang. Their careful reviews and valuable suggestions have helped me to improve the quality of my thesis. Special thanks goes to Dr. Baohua Li, who was helping me with all the studies of machine learning; I could not finish this thesis without her guidance.

I would like to thank my lab mates: Dr. Guoqing Zhou, Dr. Ning Sun, Dr. Gang Wang, Jayshree Desai, Zuoen Wang, Israel Z. Akingeneye, and Alli Ayoola Ojutiku, for the classes we had together and for all the fun we had in the past two years.

Last but not least, I would like to thank all my family for supporting and caring, not only in my studies but also my daily life

TABLE OF CONTENTS

I. Introduction

Mentally training individuals to reach their optimal performance or improve their performance even beyond their current skill levels is a desire of each sports field. For the past decades, building biofeedback systems based on brainwave patterns is the main trend to train athletes after the brain wave was found to be highly associated with the event-related period. Therefore, brain wave analysis is needed before building such a biofeedback system. Electroencephalogram (EEG) signals are popularly used brain signals that reflect the brain activity by placing electrodes on the scalp. It measures summation of electrical activities of thousands or even millions of neurons that have similar spatial orientations in the human brain (Thatcher, Biver, & M., 2004).

Golf putting is well known as a cognitive goal-directed sports action due to the continuous thinking, concentration, aiming, planning, and decision-making during the preparation period. A large amount of research showed that golf putting performance is predictable by extracting the EEG signals in specific frequency bands. Recent research proposed a training method by setting a threshold for each EEG frequency. Once the signal frequency is higher than the threshold, the feedback system would encourage the golfer to putt within 1.5 seconds (Martijn Arns MS, Michiel Kleinnijenhuis MSc, Kamran Fallahpour , & Rien Breteler , 2008). Prediction based on a threshold is propitious for applications, however, it is easy to cause misclassification due to the complicated variation of EEG signals and differences in successful patterns. With the purpose of giving an effective detection of successful pattern and precisely predicting golf performance, we proposed a classical machine learning algorithm -- Support Vector Machine (SVM) -- for analyzing EEG signals recorded during golfers' preparation periods. After the structure of this classifier is set up in the neuro-feedback system, an instructive signal would be generated to encourage golfers to make the putt if a successful pattern is detected. Otherwise, if an unsuccessful pattern is detected, it will let the golfer be more concentrated on aiming and planning.

In this thesis, we will focus on the analysis of EEG signals that is collected by the preparation period. Four features are extracted from time domain and frequency domain. Nonlinear binary SVM models have been applied. In the second chapter, EEG data extraction and feature separation are presented. In the third chapter, details of building the SVM structure and parameters optimization are presented. In the fourth chapter, all experiment results are shown with the comparison between features and SVM models.

II. EEG data analysis

A. EEG introduction

We begin with the fact that human brain weighs approximately 2.5 pounds, consumes approximately 40% to 60% of blood glucose, and consumes the same amount of oxygen as our muscles for 24 hours a day. A large amount of energy has been used to produce electricity for supporting the actions of small and large groups of neurons. Each neuron needs to constantly be recharged (Tryer,L., 1988; Niedermeyer & Silva, 2012).

EEG is the recording of electrical action by placing multiple electrodes on the scalp. It measures voltage variations caused by ionic current flows within the neurons in the brain (Niedermeyer & Silva, 2012). Scalp EEG recording shows the electrical potentials difference between two locations nearby the electrode on the scalp. However, most of the human cerebral cortex is hidden deeply beneath the scalp. It is hard to record the immediate activity from small groups of neurons, and the waveforms recorded from the electrodes on the scalp represent the cortical potential differences that come from the synchronous activity created by large groups of neurons (Tatum, Husain, Benbadis, & Kaplan, 2007).

In this chapter, we are aiming to find out the characteristics of EEG data which have a potential correlation with the golfers' performances; the method of extraction and calculation will be presented.

B. Literature review

In the last few decades, people have had increasing interest in brain waves gathered from human scalps, which is Electroencephalography (EEG) data. It has been a necessary factor in performance analysis of goal-directed sports such as shooting (Hillman, Apparies, Janelle, & Hatfield, 2000), basketball (Chuanga, Huangb, & Hunga, 2013), and golf (Muangjaroen &

Wongsawat, 2012). In addition, EEG feedback can potentially enhance the performance of those sports that need a perfect physical balance control, such as ice-skating and skiing, by improving the concentration and attention (Hammond, 2007). Particularly, golf putting is well known as a cognitive goal-directed task because of the continuous thinking, concentration, planning, and decision making during the preparation period, and it has been studied extensively in terms of EEG signals (Babiloni.C, et al., 2008). EEG-based Brain Computer Interface (BCI) systems have been developed to improve the rate of successful putts by generating a continue signal that can help a golfer decide if he is ready to give a relatively successful putt (Martijn Arns MS, Michiel Kleinnijenhuis MSc, Kamran Fallahpour , & Rien Breteler , 2008).

Generally, EEG data is divided into several bands in frequency domain: they are delta (1 – 4 Hz) theta (4 -7Hz), alpha (8 -12 Hz), beta (13 - 30 Hz), and gamma (30 -100 Hz) frequency bands. Normally, the amplitude in delta is relatively higher than other bands. It is always associated with deep sleep, and it has been applied for sleep stage research, while the Gamma band waves are shown during the short-term memory corresponding to recognized objects, sounds, or sensation (Kirmizi-Alsana, et al., 2006).Because of the weak relationship between golf putt and these two band waves, we would not consider the information in these two bands for this project. However, a large number of evidence showed that the theta band (4-7Hz), alpha band (8-12Hz) and beta band (13-30Hz, some articles separate this band into beta1 (13-20Hz) and beta2(21-30)) wave power has potential value for direct attention. Hillman et al proved the significant difference of power in theta band has showed between expert and novice rifle shooters, especially during the aiming process (Hillman, Apparies, Janelle, & Hatfield, 2000). In recent research, Lan-Ya et al. showed that relatively higher power in theta band has been found during the aiming processes for successful basketball throws (Chuanga, Huangb, & Hunga, 2013). In addition, for golf research area, Piyachatet al. proved that higher theta and alpha power are found in diverse channels in successful putt than unsuccessful putt (Muangjaroen & Wongsawat, 2012). Another research project displays that an increase in frontal-midline theta power appears in expert golfers in a golf putting task compared to novices. Therefore, the higher theta power might give a rise to occurrence of high focused attention on the performance (Baumeister, Herwegen, Liesen, & Weiss, 2007). In this case, we would only consider the information carried in 4-30 Hz, which covers theta, alpha, and beta band in the following analysis.

Beside EEG power over three frequency bands, coherence over the frequency bands is another feature in EEG-based analysis which has been used widely (Babiloni, et al., 2011; Babiloni, Brancucci, Vecchio, Arendt-Nielsen, & Chen, 2006; Rilk, Soekadar, Sauseng, & Plewnia, 2011; Davey, Victor, & Schiff, 2000). It is an extension of Pearson's correlation coefficient to complex number pairs. In EEG-based analysis, it can measure the relationship between EEG signals simultaneously recorded from two different electrode sites on the scalp at a given frequency and reflect the functional coupling among brain areas (Shaw, 1981; Babiloni, Brancucci, Vecchio, Arendt-Nielsen, & Chen, 2006). It may be more effective than PSD in inter-hemisphere analysis changes caused by cognitive tasks (Shaw, 1981). Recent research reported that a higher alpha coherence is associated with a better performance in a unimanual visuomotor task. At the same time, increase of beta1 coherence of centroparietal region and beta2 coherence of frontal region were observed (Rilk, Soekadar, Sauseng, & Plewnia, 2011). The coherences of 12 combinations of 10 electrodes in the movement period and the baseline period were calculated and the difference of these two coherences in alpha1 and alpha2 bands is relatively higher in successful putting than unsuccessful putting (Babiloni, et al., 2011).

In this chapter, amplitude in time domain, correlation in time domain, power spectrum density in frequency domain (4-30 Hz), and coherence in frequency domain (4-30Hz) will be analyzed.

C. EEG data extraction

Objects and data

26 golfers have been recruited which include expert and novice, and their age range is from 18 to 27. All golfers were asked to give 40 putts, and they were told to relax after they finished the first 20 putts. The EEG data was recorded with the movement of the club. A significant spike occurred in the club signals, and spikes are created with sudden changes in acceleration in a couple milliseconds (See Figure 2.1). Suppose the moment that spike showed is 0s; -1s – 0s is considered as the putting period that includes the club moving backward and then hitting the ball. Before -1s-0s, golfers make preparation in their minds until they start moving the putter to stroke the ball. Therefore, the EEG data from -2 second to -1 second can reveal golfers' brain states in preparation that are most correlated to their putting performance. In the following analysis, we will analyze the one second signals and try to find the correlation in different areas of the brain.

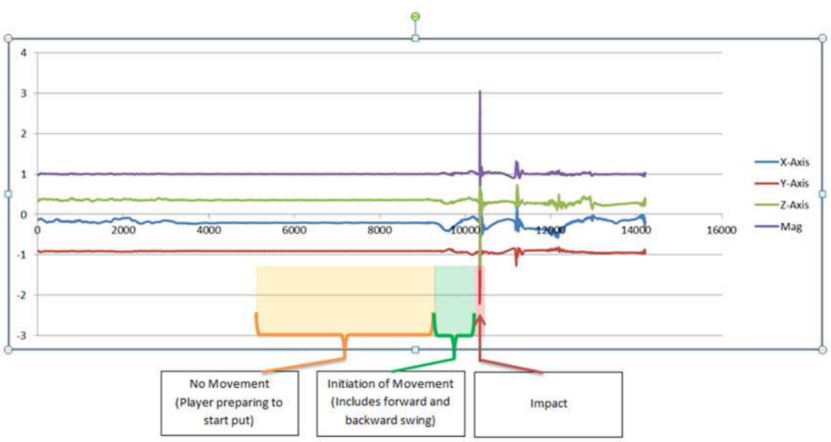


Figure2.1 club data with a spike

Golfers' putting performance is measured by two criteria. One straightforward and popular criterion is the error in centimeters which is the distance of the ball away from the hole. The other is perception of self-confidence, concentration, and quality of each stroke evaluated by golfers themselves (Crews, Martin, Hart, & Piparo, 1991). The perception has 10 grades from 1 (worst) to 10 (best). The performance of each golf putt is classified into either success or failure based on both cm errors and perception grades. In this paper, if the perception grade is higher than 8 and the cm error is less than 30 cm, we consider it a successful putt. Otherwise, it is defined as an unsuccessful or failed putt.

EMOTIV device

The device we used to collect EEG data is the 'Emotiv EEG neuroheadset,' which is made by Emotiv Company. It uses 14 sensors attached on the scalp to detect electric signals produced by the brain and transmit them wirelessly to the computer. The 14 sensors cover the locations based on the International 10-20 locations (Figure 2.2). They are: FP1, F7, F3, T3, C3, P7, P3, O1, O2, P4, C4, T4, F4, F8, and FP2. It doesn't include Fz, Cz, Pz, A1 and A2.The sampling rate for this device is 128 Hz. It is sufficient for this project since we are only interested in the 4-30 Hz EEG data. Figure 2.3 shows the device which could be attached on the scalp. After this device extracts EEG signals, they are transmitted wirelessly to the receiver which is connected with the computer.

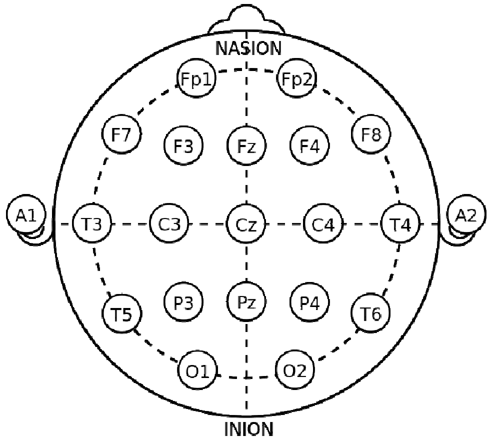


Figure2.2 International 10-20 system[1] Fig 2.3 Emotiv EPOC model 1.0[2]

We use EEG signals from 8 electrodes in the EEG headset which are placed at the left frontal, right frontal, left temporal, right temporal, left central, right central, left parietal, and right parietal areas, since these 8 areas cover the main regions for visuospatial and somatomotor processes of both the left and right brain hemispheres which are closely correlated to golf putting performance (Babiloni, et al., 2011). The corresponding 8 scalp locations are named F3, F4, T3, T4, C3, C4, P3, and P4 according to the international 10-20 scalp electrode placement system.

D. Time domain analysis

The signals we obtain directly from the receiver are time sequences with 128 samples per second, with a sampling rate of 128 Hz. Since the second to last second is considered as the target period that has a relatively high correlation with the performance, the time sequences outside this one second will be discarded in the following analysis. In this section we will discuss features in time domain that include amplitude and cross correlation.

[1]Figure2.2 International 10-20 system source: http://en.wikipedia.org/wiki/10_20_system_(EEG)#mediaviewer
[2] Fig 2.3 Emotiv EPOC model 1.0 source *ource:* http://emotiv.com/upload/media/1_big.jpg

Amplitude in time domain

Figure 3.1 (a)-(d) came from Jen (one of the golfers). Different lines represent different time sequences from 14 channels, and the amplitudes are measured with microvolts. As we can see, the alterations of 14 lines have many similarities, which is because the signals we collected from one electrode not only come from the neurons right beneath the electrode, but are also mixed with the signals from neurons beneath other electrodes.

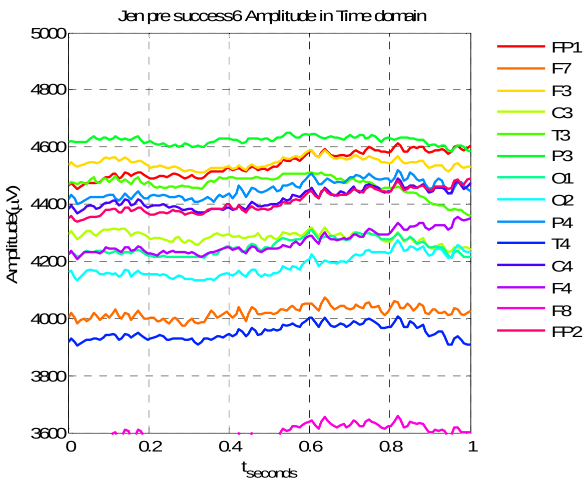


Figure2.4 (a) Quality: 10 CM Error: 0

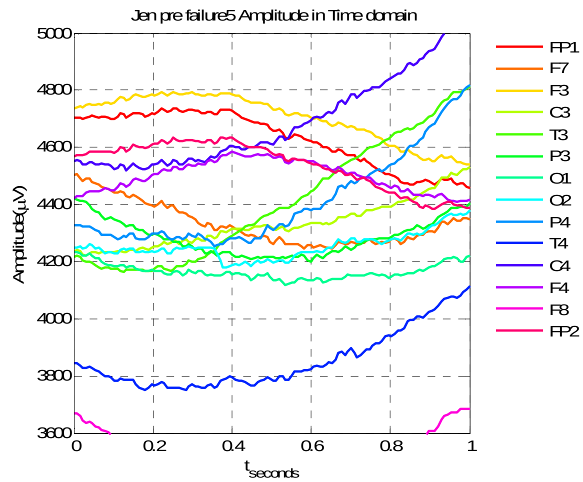


Figure2.4 (b) Quality: 10 CM Error: 47

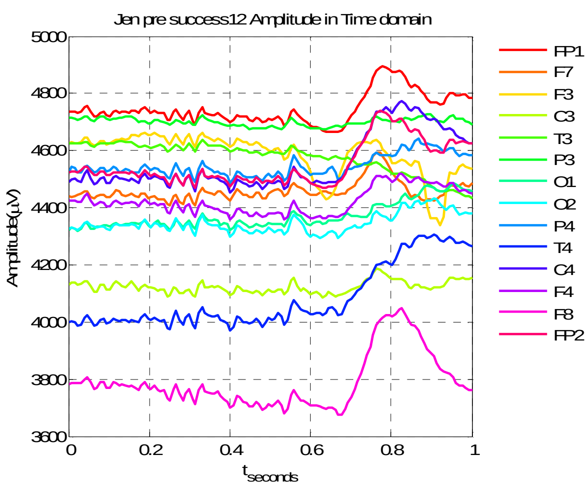


Figure2.4 (c) Quality: 10 CM Error: 0

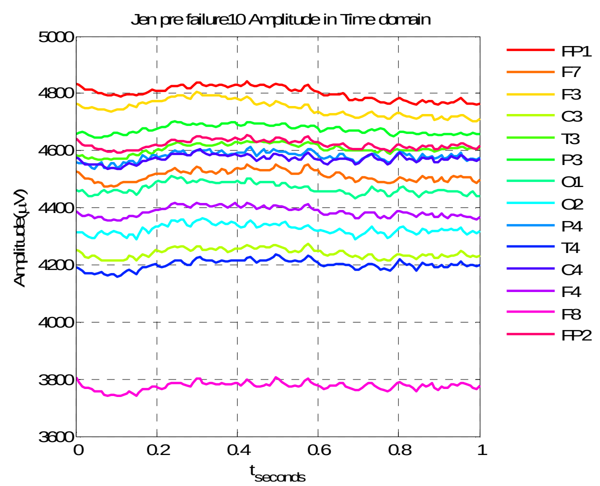


Figure2.4 (d) Quality: 10 CM Error: 86

Qualities of both Figure2.4 (a) and (c) are 10, and cm errors are 0. However, the brain waves are totally different. For those successful putts that have different qualities and cm errors, the differences are even more indistinguishable.

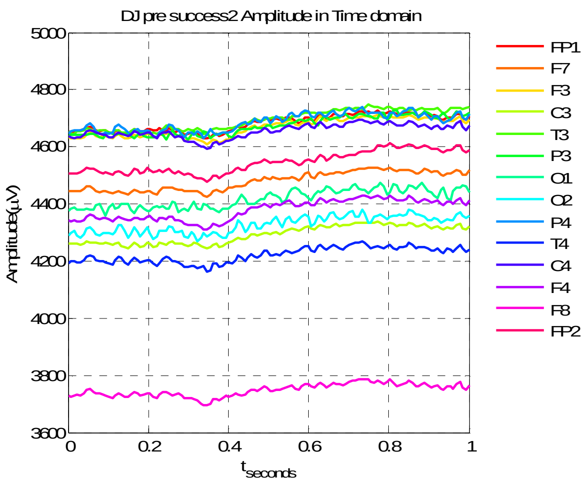


Figure2.5 (a) Quality: 10 CM Error: 0

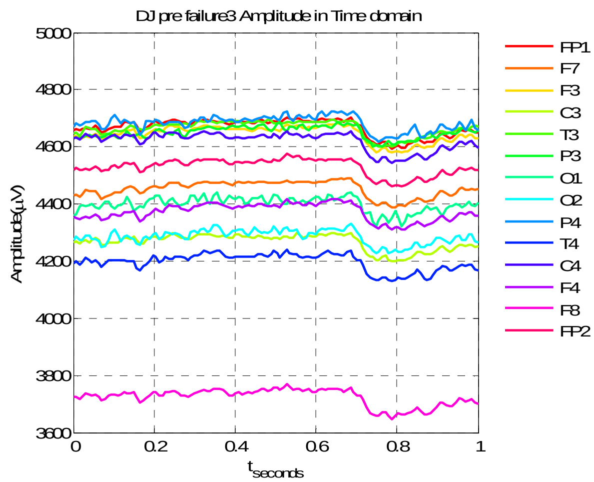


Figure2.5 (b) Quality: 8 CM Error: 69

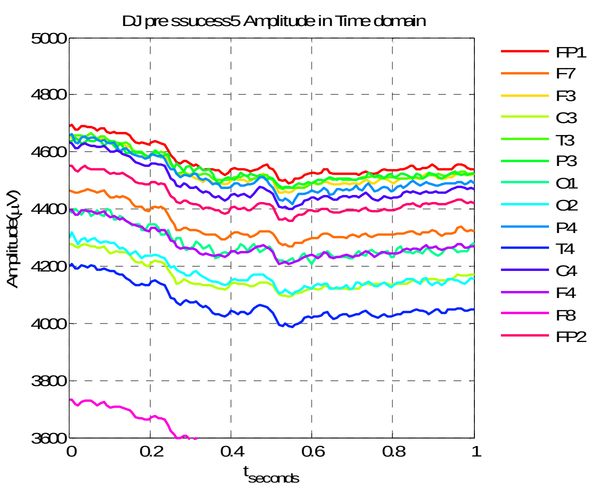


Figure2.5(c) Quality: 10 CM Error: 0

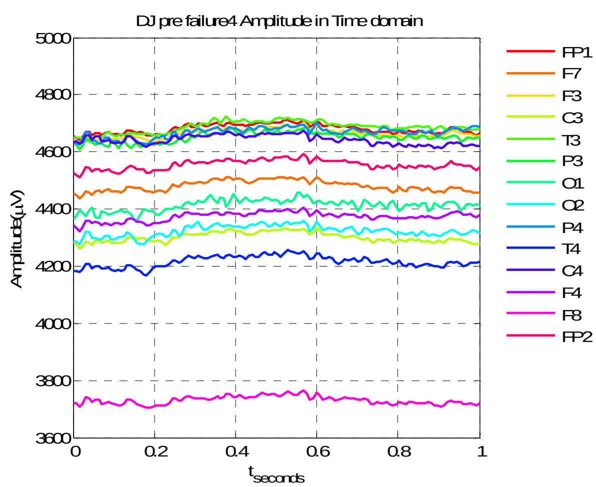


Figure2.5 (d) Quality: 8 CM Error: 77

Figure 2.5 (a)-(f) came from DJ (one of the golfers), and the signals from him are more stable than Jen. However, it is still hard to distinguish successful and failed putts by observing the signals from the amplitude in time domain. This situation also happened to other golfers' signal, but we won't show all the figures here. Because of the insignificant characteristic of time sequence amplitude, we do not consider it as a feature to classify the data. Another reason is that if we used all the time sequences, it would give 1792 amplitude values for every trials making the calculation cost too high. From the perspective of building an instantaneous feedback system, the high dimension trial calculation would cost a time lag which may provide false instruction to the golfer.

Cross correlation

Another feature that has been widely used in EEG time domain analysis and signal processing is cross correlation. For example, Bahcivan et al. used cross-correlation to prove the existence of common activity of two different locations during the epileptic seizures at a particular band (Bahcivan, Hopkins, Zhang, Mirski, & Sherman, 2001) . In addition, Hermanto suggested that cross-correlation is an important reference to measure the similarity of EEG signals that could be used to classify features in the brain computer interface signal processing (Hermanto, Mengko, Indrayanto, & Prihatmanto, 2013).

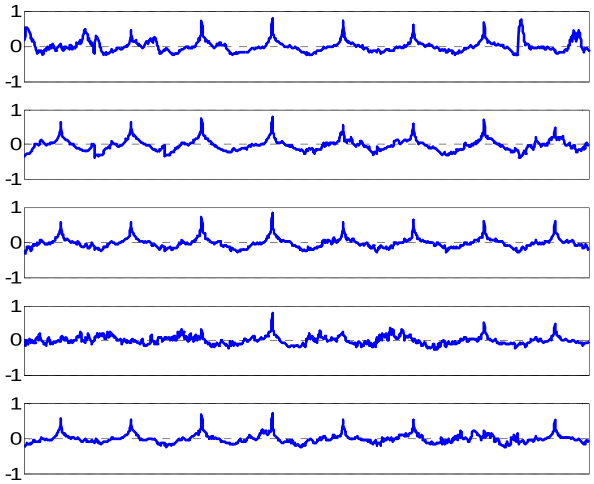
In this project, one hypothesis is that the time sequences in successful golf putts have similar patterns. At the same time, there exists a relative big difference between successful golf putts and unsuccessful ones. If the hypothesis is supported by the real data, which means that the cross correlation between successful conditions would be higher than between successful and unsuccessful conditions. The cross correlation would be a reasonable feature that could be used in prediction by computing it between unknown EEG data and known successful time sequences.

Cross-correlation has been defined by the function:

$$r_{xy}[k] = \sum_{n=-\infty}^{\infty} x[n]y[n-k]\,, k = 0\,, \pm 1\,, \pm 2, \ldots \qquad 1.1$$

After this introduction of definition of cross correlation, we need to decide which two signals would give the significant difference for support vector machine training. What we did is to calculate cross correlation between every trial with 10 successful trials. With the purpose of including all the successful features from the same person and some successful features from other persons, if there are no 10 successful trials for this person, the 10 successful trials with the closest distance to the target trial are selected. The order in the data set is fixed, so we searched forward to get five successful trials and backward to get another five successful trials. After we separately calculated the cross correlation between the target trial and reference trials, the middle areas were extracted from the long correlation sequence. For example, both the target time sequence and reference sequence have 128 values, so the total cross-correlation sequence has $2 * 128 - 1 = 255$ values. We only used the middle part of average cross correlation sequence, which is the fiftieth to the two hundredth values, for the consideration of dimensional reduction and sufficient overlaps between two signals. For the convenience of viewing, we combined 8 channels together.

Figure 2.6 (a) represents 5 cross-correlation sequences that come from 5 successful putting of one golfer and figure 2.6 (b) comes from the unsuccessful putts. Five curves in the left figure have significant 8 peaks during 8 channels, only the fourth one is irregular. This means that most of the successful trial has correlation with other successful trials with the fourth one showing no correlation. In contrast, the five curves in the right figure are all irregular except the second one which gives the evidence that there are no marked correlations between successful and unsuccessful putting.



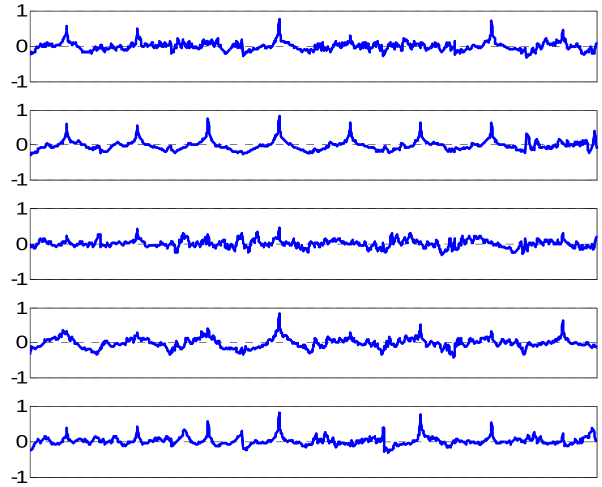
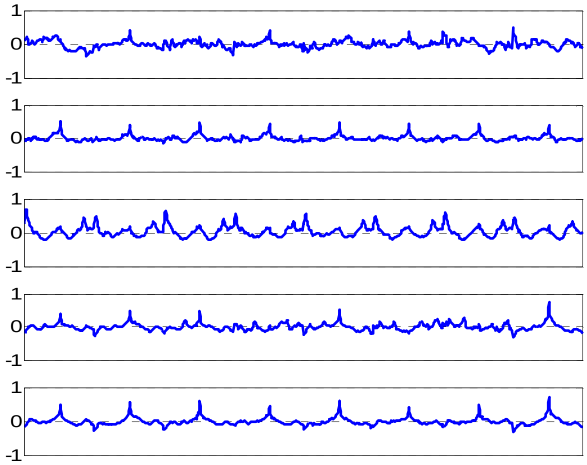


Figure 2.6 (a) successful CC   Figure 2.6 (b) unsuccessful CC

However, not all the cross-correlation curves from the golfers showed significant difference. Figure 2.7 (a) - (b) came from another person. The second, third and fourth curves in the right figure have no spike in all 8 channels. Although, the number of peaks in the five curves obtained from cross- correlation between successful putting and reference putting is not as significant as those in figure 2.6 (a), they show more similarities than the correlation between unsuccessful putting and reference putting.



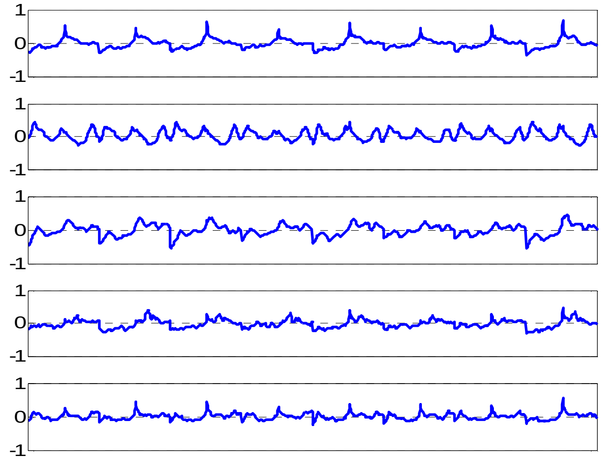


Figure 2.7(a) DANI successful CC   Figure 2.7(b) DANI unsuccessful CC

Because of this characteristic, the cross-correlation sequences between target trials and reference successful trials are good features as the input of Support Vector Machine.

E. Frequency domain analysis

Power spectrum density analysis

Fast Fourier Transform (FFT) is an algorithm to calculate the discrete Fourier transform (DFT). A Fourier Transform converts time domain signals to frequency domain signals; FFT is very fast at calculating such transformations.

The function below shows the Discrete Fourier Transform:

$$X(k) = \sum_{n=1}^{N} x[n]e^{-j\omega n} \qquad 2.2$$

while

$$\omega = \omega_k = \frac{2\pi}{N}k \qquad 2.3$$

Power spectral density (PSD) describes how the power of a signal or time series is distributed over the different frequencies. For discrete time signals, the definition of the power spectral density can be defined as:

$$s_{xx} = \frac{(\Delta t)^2}{T}\left|\sum_{n=1}^{N} x[n]e^{-j\omega n}\right|^2$$

$$= \frac{\Delta t}{N} \left|\sum_{n=1}^{N} x[n]e^{-j\omega n}\right|^2$$

$$= \frac{1}{(F_s * N)} \left|\sum_{n=1}^{N} x[n]e^{-j\omega n}\right|^2$$

$$= \frac{1}{(F_s * N)} |X(k)|^2 \qquad 2.4$$

while $x[n] = x(n * \Delta t)$, T = N*Δt , $F_S = \Delta t$.

The function 2.4 would be used in Matlab coding. The default window function used in FFT is the rectangular window. For the sake of avoiding power leakage, a popular window function, hamming window, has been applied. The function of the hamming window is

$$w(n) = 0.54 - 0.46\cos\left(2\pi \frac{n}{N}\right) \qquad 2.5$$

Coherence

As mentioned previously, a large amount of research used coherence as a feature to analyze EEG signals and the relationship with human behavior. For the same reason, we would consider coherence to measure the quantity of phase constancy between two signals. If the relationship between two signals is constant, then the coherence is 1. If the relationship between two signals changed randomly, then the coherence is 0.

Coherence in signal analysis is defined as:

$$Coh\,(f) = \frac{|S_{XY}(f)|^2}{S_{XX}(f)S_{YY}(f)} \qquad 2.6$$

two time sequences are denoted as X and Y, $S_{XX}(f)$ and $S_{YY}(f)$ are the auto spectrum estimate of X and Y at a given frequency f, respectively, and $S_{XY}(f)$ is the cross spectrum estimate of these two time sequences. For the purpose of our experiment, we need spectral coherences between 4Hz and 30Hz of some combinations of 1-second EEG trials from the 8 scalp locations which are F3, F4, T3, T4, C3, C4, P3, and P4. To capture subtle and significant variations of the EEG patterns representing golf putting mental preparation state, we consider 22 pairs of these 8 scalp locations. To fully evaluate the inter-hemispheric functional coupling, all 16 combinations of 4 left scalp locations and 4 right scalp locations are included, which are F3-F4, T3-T4, C3-C4,

P3-P4, F3-T4, F3-C4, F3-P4, T3-F4, T3-C4, T3-P4, C3-F4, C3-T4, C3-P4, P3-F4, P3-T4, P3-C4. Since frontal areas are closely associated with planning which is the process of thinking and organizing the activities required to achieve a desired goal, they play an important role in golf putting mental preparation. Therefore, the frontal intra-hemispheric functional coupling with other cortical regions in the same hemisphere are particularly evaluated (Babiloni, et al., 2011). Here, 6 combinations of left and right frontal areas with the respective other three areas in the same hemispheres are included, which are F3-T3, F3-C3, F3-P3, F4-T4, F4-C4, F4-P4. Based on 1-second EEG trials extracted in the last subsection, the spectral coherences from 4Hz to 30Hz of the 22-pair electrodes can be computed. These 22 spectral coherences are concatenated into a vector which will be an input of the classification algorithm in the next step.

III. Applied SVM on EEG data

A. SVM introduction

In the previous chapter, we proposed 1) cross-correlation between target trials and the 10 closest reference trials; 2) power spectrum density in selected 8 channels over 4-30Hz ; 3) average power spectrum over theta (4-7Hz), alpha (8-12Hz), beta1 (13-20Hz) and beta2 (21-30Hz) band; 4) all coherences between 4Hz and 30Hz of 22 combinations for 8 electrodes on the scalp 5) average coherence in 4 bands that determined in the same manner as average PSD bands, used as the features to fully characterize the EEG patterns during putting preparation. However, it is unlikely to distinguish successful and unsuccessful EEG pattern directly. In order to effectively predict the putting performance, we choose support vector machine (SVM) algorithm to classify the all feature of EEG patterns that listed above into two categories corresponding to successful and unsuccessful putts since this algorithm has rigorous formulation and has been used in many EEG-based pattern recognition applications

B. Literature review

SVM was first introduced by (Bernhard E. Boser, Isabelle M. Guyon, & Vladimir N. Vapnik, 1992) then it has been widely used in data analysis such as classification and pattern recognition applications (Schlkopf & Smola, 2002) (Shen, Li, Ong, Shao, & Wilder-Smith, 2008) (Li, Zhang, & Du, 2013) (Parvez & Paul, 2014). Paul claimed that decision patterns showed more potentials compare to no choice brain pattern which makes it possible to predict the decision-making EEG signals. (Paul, Leung, Peterson, Sejnowski, & Poizner, 2010). In another research (Li, Zhang, & Du, 2013), SVM is used to classify six different movement patterns in EEG signals, participants were asked to imagine different sports that were related to still, walking, squatting and stand up, up to the slopes, down to slopes and running. The result gives a relatively high prediction accuracy that reached 78.9%, this result proved that SVM could effectively classify EEG on complicated thinking. Besides those, it is widely used in clinical areas such as Eplieptic (Parvez & Paul, 2014) and fatigue measurement (Shen, Li, Ong, Shao, & Wilder-Smith, 2008), furthermore, SVM has successfully classify the EEG data from alcoholics and non-alcoholics , the test accuracy reached 94.67% (Kousarrizi, Biomed. Eng. Dept., Ghanbari, Gharaviri, & Teshnehlab, 2009)

C. Binary SVM classifier

Basic concepts of SVM

For Binary SVM classifier, the input data consists of two labels corresponding to two classes of data, we call them positive examples and negative examples. SVMs represent those data as points in the high dimension space, then determining an optimal separating hyperlane in the space to classify those examples. (Drish, 1998) We uses given training examples $x_i \in R^n, i = 1,2 \ldots\ldots, l$, labeled by $y_i \in \{+1, -1\}$, to create $f(x)$ by optimizing one or more parameters. The decision function $f(x)$ can be used to predict the label of any test examples. (Schlkopf & Smola, 2002)

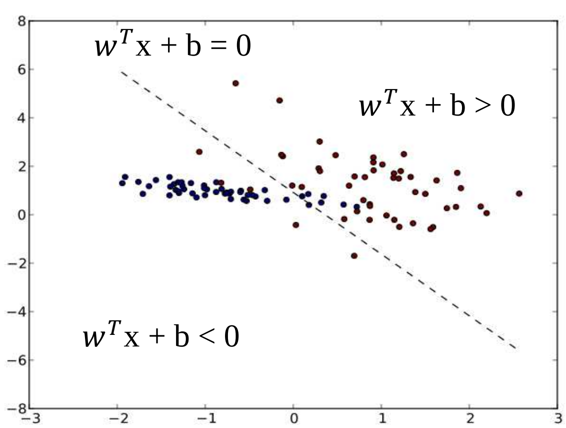


Fig3.1 Scatterplot of a binary classification dataset[3]

Figure3.3 is an example for linear binary dataset, the decision function is x + b, w is the weight vector and b is called bias. The hyperlane

3.1

divides (the dash line in the middle of dots in Figure) the dots into two: dots above the line are called positive examples, dots under the line are called negative examples.

a) Margin and optimization problem

It is easy to find lots of hyperlanes separating the training example . To decide the optimal unique hyperlane among all the hyperlanes, margins of separation between any examples are the important standards. The optimal hyperlane has the maximum margin. (See Fig 3.4)

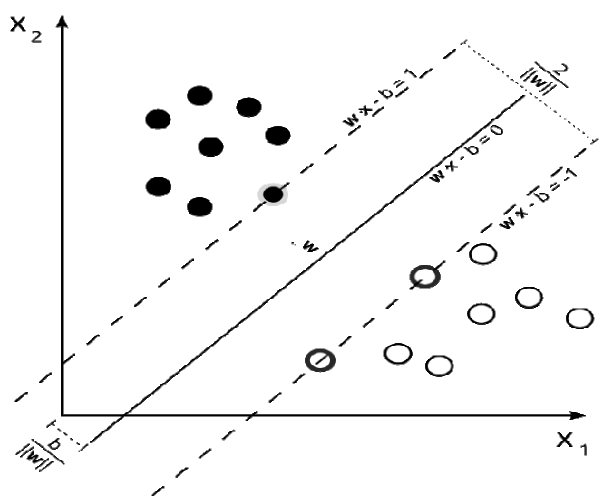


Fig 3.2 Graphic showing the maximum separating hyperlane and the margin.[4]

Mathematically, it is decided by the solution of

3.2

[3] Linear-svm-scatterplot. N.d. Wikipedia. Web. 22 Oct. 2013. <http://en.wikipedia.org/wiki/File:Linear-svm-scatterplot.svg>.

[4] *Svm Max Sep Hyperplane with Margin*. N.d. *Wikipedia*. Web. 16 Feb. 2008. http://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png

Compare to Figure 3.1, Figure 3.2 has an optimal hyperlane for the same examples. As we can see the distance between the closest point and the point is $1/\|w\|$, if we consider both sides, the distance would be $2/\|w\|$. To maximize $2/\|w\|$ is to minimize $\|w\|/2$, so the optimization function is

$$\min_{w,b} \frac{1}{2}\|w\|^2 \qquad 3.3$$

Subject to: $y_i(w \cdot x_i + b) \geq 1$ 3.4

The reason we put the ≥ 1 instead of ≥ 0 is because no matter what kind of < w, b> we find from this problem, we can find another < w', b'>, which multiplied by $\lambda(0 < \lambda < 1)$, that can reach a minimizer $\frac{1}{2}\|w\|^2$, putting ≥ 1 on the right side effectively fixed this problem. Furthermore, $y_i(w \cdot x_i + b)$ make sure that $w \cdot x_i + b = +1$ when $y_i = +1$, $w \cdot x_i + b = -1$ when $y_i = -1$.

To solve this problem, we can introduce the Lagrange multipliers $\lambda_i \geq 0$ and the Lagrangian:

$$L(w, b, \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \lambda_i \left[y_i(w \cdot x_i + b) - 1\right] \qquad 3.5$$

Take the partition differential of L respect to w, b

$$\frac{d}{dw} L(w, b, \lambda) = w - \sum_{i=1}^{m} \lambda_i y_i x_i = 0 \qquad 3.6$$

Then

$$w = \sum_{i=1}^{m} \lambda_i y_i x_i \qquad 3.7$$

$$\frac{d}{db}L(w,b,\lambda) = \sum_{i=1}^{m} \lambda_i\, y_i = 0 \qquad 3.8$$

Let's plug the 3.7 and 3.8 in 3.5 we can get the dual problem

$$\max_{\lambda} \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \lambda_i\, \lambda_j\, y_i y_j x_i^T x_j \qquad 3.9$$

Subject to

$$0 \le \lambda_i \le C \;\; for\; i = 1,2,\dots,n \qquad 3.10$$

Once we get λ , it is easily to obtain w by equation 1.5. The decision function $f(x)$ would be as follow

$$f(x) = sgn(\sum_{i=1}^{m} \lambda_i\, y_i x_i^T x + b) \qquad 3.11$$

The x in equation 3.11 is the data that needs to be classified. If *f(x) = 1*, the test data x would be considered as positive samples. If *f(x) = -1*, it would be consider as negative samples.

b) Kernel functions and non-linear classification

For some complicated situation, it is hard to generate a liner hyperlane which separates positive and negative examples perfectly. (Figure1.10) when positive and negative examples have overlaps, non-linear classifier is more efficient.

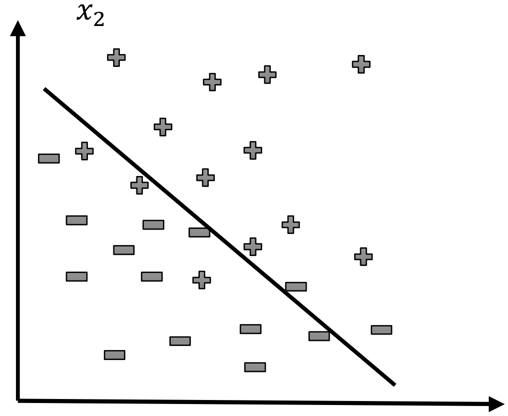


Figure 3.3（a）linear classifier

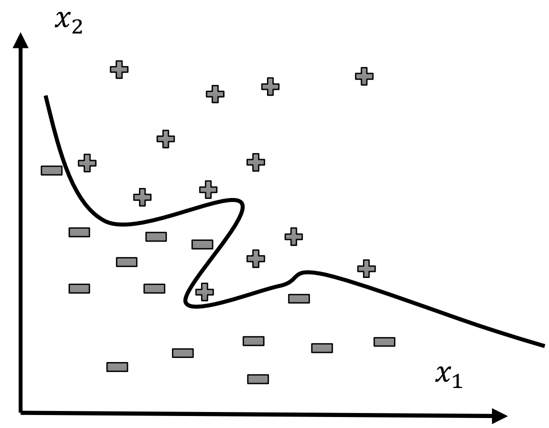


Figure 3.3(b) non-linear classifier

To generate a non-linear hyperlane, we need to replace the $x_i^T x_j$ in function 3.9 by non-linear kernel functions such as polynomial kernels, Gaussian radials basis kernels and sigmoid kernels. Gaussian radial basis function（RBF）was selected, it is defined as:

$$K(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right), \gamma > 0 \qquad 3.12$$

If we generate a matrix with K with data point $x_1, x_2, \dots \dots . x_m$ the $K(x_i, x_j)$ is positive definite, then it is possible to generate a Φ function such that

$$K(x_i, x_j) = < \Phi(x_i), \Phi(x_j) > \qquad 3.13$$

$$f(x) = sgn(\sum_{i=1}^{m} \lambda_i y_i \Phi(x_i)^T \Phi(x) + b) \qquad 3.14$$

Then the final decision function would be written as 3.14 (Schlkopf & Smola, 2002)

c) Important parameters

To construct a SVM model with a higher classification accuracy, both parameters C and γ need to be tuned very carefully. In our study, we use iterative grid search and cross validation techniques to optimize C and γ. Usually, in the experiment, all available samples can be partitioned into two parts. One is used to find best C and γ and construct SVM model and it is called training data. The other is used to test the performance of well-trained SVM model and so called testing data. Now, we focus on training data to optimize the C and γ and construct SVM model. To find better C and γ and at the same time reduce the computation complexity, an iterative grid search with exponentially growing values of two parameters in (Hsu, Chang, & Lin, 2010) is used for our problem. At the first search, the values of C and γ are bounded in the coarse sets $\{2^{-5}, 2^{-5+h}, \dots, 2^{-5+19h}, 2^{-5+20h}\}$ and $\{2^{-10}, 2^{-10+h}, \dots, 2^{-10+14h}, 2^{-10+15h}\}$, respectively, where the step h to discretize the power of 2 is 1. Given any pair of C and γ, a 5-fold cross validation is performed, where the training data are divided into 5 subsets of equal size. In the cross validation, any four of these five training subsets are used to train SVM model and the remaining is used to test this model. Thus, each sample of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified. We pick the best C and γ in the two ranges with the highest cross validation accuracy. In addition, we add a bound of the average training accuracy based on any four training subsets in the cross validation. If this training accuracy of one pair of C and γ is lower than the bound, then this pair has to be eliminated from the parameter ranges to further guarantee good SVM model and save the computation cost. In our experiment, such a training accuracy bound is set to 80%. Suppose the best C and γ to be 2^i and 2^j at the first search. Then, a smaller region with the finer grid at the second search can be identified as $\{2^{i-10h}, 2^{i-9h}, \dots 2^{i+9h}, 2^{i+10h}\}$ and $\{2^{j-10h}, 2^{j-9h}, \dots 2^{j+9h}, 2^{j+10h}\}$, where the new step h is 0.5. The same procedure can be followed to find the best C and γ in the new ranges. In our method, we will implement the grid search 5 times. To iteratively improve C and γ, we narrow down their ranges by shrinking step h

to $(0.5)^2$, $(0.5)^3$, and $(0.5)^4$ for the third, fourth and fifth search, respectively. Using the best C and γ after five searches, the final SVM model is well trained based on the whole training set and also the corresponding training accuracy can be computed.

IV. Experiment result

To demonstrate the performance of the proposed method, we will use support vector machine with RBF kernels to analyze 573 1-second EEG trials extracted by the procedure given in the previous section and the corresponding putting performance data. In our method, the feature as the input of SVM is the spectral coherence vector defined in the previous section. We will compare prediction accuracies using this feature with those using cross correlation, PSDs from 4Hz to 30Hz, coherence, and average coherence over theta, alpha, beta1 and beta2 bands to show the proposed feature is better than the other three commonly used features.

In our experiments, 573 EEG trials were recorded from 26 golfers. Unlike (Muangjaroen & Wongsawat, 2012) (Martijn Arns MS, Michiel Kleinnijenhuis MSc, Kamran Fallahpour , & Rien Breteler , 2008; P. Terry, P. Mahoney, 2006; Baumeister, Herwegen, Liesen, & Weiss, 2007; DJ & DM., 1993; Babiloni.C, et al., 2008) where the golf putting skill levels and ages are strictly controlled, some of them are excellent experts and some are novices who may be playing golf for the first time. The participants' putting skill varies very much. Their ages also spread widely from 18 to 71 years old. These may make our EEG pattern recognition and putting performance prediction much more difficult. However, the experimental results still clearly show that the proposed method outperforms the SVM with cross-correlation, PSDs, average PSD and average coherence as inputs.

To implement our method, we divided 573 samples into training data and testing data. We made three different partitions. At each partition, two third of the total samples are used as training data and the other one third is used as testing data. That is to say, there are 383 training data and 190 testing data. The training data includes 196 successful trials and 187 unsuccessful trials and the testing data includes 97 successful trials and 93 trials. Since the sampling rate of the recorded EEG signal is 128Hz, the PSDs of each EEG channel and the spectral coherences of each pair of EEG channels were computed at 65 frequency bins from 0Hz to 64Hz. We only need PSDs of the 8 channels and the spectral coherences of 22-pair channels from 4Hz to 30Hz. We concatenated PSDs of 8 channels and the spectral coherences of 22-pair channels to form a 216-dimensional PSD vector and a 594-dimensional coherence vector which will be used as inputs of the SVM. We also took the averages of PSDs and spectral coherences over theta, alpha, beta1 and beta2 bands to compact their carried information. Then we concatenated these numbers across the 8 channels to form a 32-dimensional average PSD vector and across the 22-pair channels to form a 88-dimensional average coherence vector as inputs of the SVM. We will compare the performance of the SVM with four types of inputs in this paper.

The confusion matrix structure is like the following table:

Table 1 Confusion matrix example

Confusion matrix example		Actual label	
		successful	unsuccessful
Predict Result	successful	165	31
	unsuccessful	22	165

Used the first PSD training accuracy as an example

Before using the method we proposed to verify the data set of golf related EEG signal features, we firstly applied this algorithm on the data set of two direction thinking EEG signals. A person is asked to wear the Emotiv device and think about the direction of forward and backward. 100 samples have been collected with 50 forward thinking EEG data and 50 backward thinking EEG data. Among the 100 samples, 68(close to 2/3 of all data) of them are used as training and 32 of them used as test data. The results showed that 13 backward and 15 forward have been detected, which gives a test accuracy of 87.5%.

We did 21 tests of each features, the full results are presented in Appendix A. For each tests, we changed the combinations of test trials and training trials with fixed successful/unsuccessful ratio.

The combinations were randomly generated with the seed changed. Four test results are given in the following 4 tables, each table gives 1 best result of one feature from the 21 tests corresponding to 3 results of other 3 features, and four results listed in the same table share the same seed.

Table 2 Highest cross-correlation with other features

Features Random generator V5	**Optimal parameters (C, γ)**	**Cross-validation accuracy (%)**	**Training accuracy (%) confusion matrix**	**Testing accuracy (%) confusion matrix**
Cross Correlation (4-30Hz), seed [71]	C = 7.0250 γ = 0.0653	53.4211	99.48 Confusion matrix =[194 2 0 187]	**<u>58.95</u>** Confusion matrix =[67 30 48 45]
PSD (4-30Hz), seed [71]	C = 2.896e+03 γ = 30.6433	47.1053	80.16 Confusion matrix =[171 25 51 136]	50 Confusion matrix =[56 41 54 39]
Coherence (4-30Hz), seed [71]	C = 1 γ = 1.1388	58.6842	93.21 Confusion matrix =[184 12 14 173]	52.63 Confusion matrix =[36 61 29 64]
Average Coherence seed [71]	C = 0.4 γ = 12.3377	59.7368	78.85 Confusion matrix =[147 49 32 155]	52.63 Confusion matrix =[33 64 26 67]

The highest overall test accuracy of cross-correlation is 58.95% (see table 2), 63 successful patterns have been detected from 97 successful patterns, which gives a 65% successful detection accuracy. On the contrast, the overall test accuracy of PSD and coherence and average coherence are relatively lower than cross-correlation over this combination of training and test data set. However, the unsuccessful detection accuracy reaches 67/(67+26)=72% in average coherence and 64/(67+29)=68.8% coherence. This situation may be caused by inconspicuous of successful coherence and average coherence patterns in the training set, therefore, the classifier is trend to target the unknown trials to decision of unsuccessful putting.

Table 3 Highest power spectrum density with other features

Features Random generator V5	**Optimal parameters (C, γ)**	**Cross-validation accuracy (%)**	**Training accuracy (%) confusion matrix**	**Testing accuracy (%) confusion matrix**
Cross Correlation (4-30Hz), seed [1]	C = 4 γ = 0.0089	57.1053	84.07 Confusion matrix =[178 18 43 44]	57.89 Confusion matrix = [63 34 46 47]
PSD (4-30Hz), seed [1]	C = 8192 γ = 28.1	52.3684	86.16 Confusion matrix =[165 31 22 165]	**53.68** Confusion matrix = [51 46 42 51]
Coherence (4-30Hz), seed [1]	C = 3.0048e+04 γ = 0.001	60	86.42 Confusion matrix =[171 25 27 160]	54.74 Confusion matrix = [52 45 41 52]
Average coherence seed [1]	C = 609 γ = 1.0443	57.8947	99.74 Confusion matrix =[195 1 0 187]	54.21 Confusion matrix = [55 42 45 48]

The highest overall test accuracy gives the percentage of 53.68 (see table 3) which is lower than other features. Except these combinations, the complete results (Appendix A, Table B) show that most of the result that calculated with PSD features gives relative lower overall test accuracy than other features. It gives the efficient evidence that using power spectrum density directly is more possible to give a poor performance in classification. In this case, we may use other features to training the final SVM model in the feature.

Table 4 Highest coherence with other features

Features Random generator V5	**Optimal parameters (C, γ)**	**Cross-validation accuracy (%)**	**Training accuracy (%) confusion matrix**	**Testing accuracy (%) confusion matrix**
Cross Correlation (4-30Hz), seed [81]	C = 61.2866 γ = 0.0010	58.4211	80.94 Confusion matrix =[173 23 50 137]	54.74 Confusion matrix =[55 42 44 49]
PSD (4-30Hz), seed [81]	C = 6049 γ = 9.5137	52.8947	80.68 Confusion matrix =[151 45 29 158]	43.16 Confusion matrix =[45 52 56 37]
Coherence (4-30Hz), seed [81]	C = 2 γ = 1.834	55.2632	99.22 Confusion matrix =[194 2 1 186]	**62.11** Confusion matrix =[56 41 31 62]
Average Coherence seed [81]	C = 25268 γ = 0.015	56.0526	80.68 Confusion matrix =[160 36 38 149]	60 Confusion matrix =[62 35 41 52]

The coherence gives the highest overall test accuracy among all the features and all the test, 62.11% (see table 4). At the same time, average coherence gives an overall test accuracy - 60% with the same training and test data set, especially the successful detection accuracy of average coherence is 64%. This result illustrate that using coherence and average coherence as features is more possible to give a good performance of prediction in this situation , and the training data set of this combination have markedly characteristic successful patterns and unsuccessful patterns. On the contrast, the overall test accuracy of PSD is only 43.1 which proved the previous hypothesis – using PSD as features to classify the golfer's performance gives high degree of misclassification.

Table 5 as following is another example to prove the superiority of using coherence and average coherence as features.

Table 5 Highest average coherence with other features

Features Random generator V5	**Optimal parameters (C, γ)**	**Cross-validation accuracy (%)**	**Training accuracy (%) confusion matrix**	**Testing accuracy (%) confusion matrix**
Cross Correlation (4-30Hz), seed [151]	C = 12.8839 γ = 0.0049	62.3684	87.47 Confusion matrix =[181 15 33 154]	51.58 Confusion matrix =[54 43 49 44]
PSD (4-30Hz), seed [151]	C = 25268 γ = 20.7494	51.3158	89.06 Confusion matrix =[175 21 21 166]	49.47 Confusion matrix =[53 44 52 41]
Coherence (4-30Hz), seed [151]	C = 1.5024 γ = 0.0014	55.7895	84.33 Confusion matrix =[167 29 31 156]	53.16 Confusion matrix =[54 43 46 47]
Average Coherence seed [151]	C = 70 γ = 0.4585	58.1579	80.68 Confusion matrix =[161 35 39 148]	**<u>60</u>** Confusion matrix =[54 43 33 60]

Although the variations of training and testing data sets caused by the random selection have strong influence over the structure building and classification, the parameters and structures corresponding to the highest successful detection accuracy or failure detection accuracy would be used in the final BCI system.

From all the results we obtained, we claim that coherence has significant high overall test accuracy among all the test and features, the accuracy of average coherence is not worse than coherence in most of the cases. However, because of the compression of information, average coherence took shorter time to selecting the optimal C and γ corresponding to cross-validation. This is the trade-off between test accuracy and computation cost.

V. Conclusion and discussion

This thesis has proposed a method to predict the putting performance of a golfer based on her/his EEG signals. The method collects multi-channel 1-second EEG signals before the actual putting action, extracts features from the EEG signals, and analyzes the EEG signals with SVM to predict the outcome of the putting. The operations can be implemented in a BCI system, which can help a golfer to improve her/his putting performance by providing positive feedback (such as a tone) when the EEG signals indicate a high chance of successful putting.

Four features have been used, and they are: cross correlation, PSD, coherence and average coherence. Experiment results indicate that using coherence has the highest accuracy, followed by cross correlation, average coherence, and PSD.

All results show that the training accuracies are much higher than testing accuracies for all four features. The main reason is that the hyperlanes that are used to separate the successful and failed trials overfitted the training data sets, especially in the high dimensions classifications. Overfitting training data set results in a perfect separation of the training examples. However, it might not work properly for new data samples. Generally, Overfitting occurs when positive and negative examples are indistinguishable.

In the experiment, we randomly selected the training data with a fixed ratio of all the available data, and the result changed with the variation of training data combinations. Comparing all the 21 combinations, we found that some combinations yield higher test accuracies due to a better separation between the successful and failed trials in the training sets; other combinations give relatively low test accuracy. According to this phenomenon, we know that some of the successful trails do not have marked patterns, and using those data as training data would increase the degree of misclassifications. However, since it is possible to select the training data set, more selections and tests are needed to optimize the training data composition.

In addition, it has been claimed that the coherence in theta alpha and beta band changed significantly with age (Vysata, et al., 2014; Vysata, et al., 2014). This is an important factor that causes the misclassification since the age range of participants in this project is from 18 to71. In addition, according to the Neurofeedback training research (Martijn Arns MS, Michiel Kleinnijenhuis MSc, Kamran Fallahpour , & Rien Breteler , 2008), the successful power spectrum patterns in theta, alpha , beta1 and beta2 vary from different persons. To enhance the accuracy under this individual diversity, more training data is needed to build personalized training data set in order to distinguish the different successful putting patterns. In the future, this proposed method would be separately applied on expert and novice individuals which would give rise to a higher predicition accuracy.

To further improve the prediction accuracy, we can consider to enhancement to the SVM implemented in this thesis.

Probabilistic outputs of SVM

The SVM makes binary decisions based on the input data, and there is a high chance of misclassifications, even for well trained SVM structures. To improve the prediction accuracy, probabilistic outputs would be more reasonable than a simple binary decision. In 1999, Platt proposed a method to approximate the posterior probability by map the binary decision to a sigmoid function (Platt, 1999)

$$\Pr(y = 1|x) = \frac{1}{1 + \exp\,(Af(x) + B)}$$

where *f(x)* in this function is the decision that made by the SVM, A and B are parameters that are obtained by the following maximum likelihood problem (Lin, Lin, & Weng, 2007)

$$\min_{A,B} - \sum_{i=1}^{l} (t_i \log p_i + (1 - t_i) \log(1 - p_i))$$

$$t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{if } y_i = +1 \\ \frac{1}{N_- + 2} & \text{if } y_i = -1 \end{cases} \quad i=1,2,3,\dots l$$

where N_+ is the number of positive examples and N_- is the number of negative examples. However, since this probablistic method is based on the binary classification reslut, in another word, it is a posterior probabily that maps the original binary classifucation reusult to a probility. So that the probabilty is influenced by original classification result. In addition, the ratio of positive examples and negative examples strongly affects the probabilistic result. Consequently, if we want to use this method, we need to avoid quantities imbalance of positive and negative trials. Otherwise, it would directly map all the test data to one group (either positive and negative) with higher ratio in the training data set.

Muli-class SVM

Another popular transformation of binary SVM is multi-class SVM, which is capable to map the high dimension examples to more than two categories (Schlkopf & Smola, 2002). In the future work, the EEG features that have been used above would be separated in 3 groups corresponding to successful, normal and unsuccessful. All the trials without significant successful and unsuccessful characteristic would be classified in the normal group.

References

Babiloni, C., Brancucci, A., Vecchio, F., Arendt-Nielsen, L., & Chen, A. C. (2006). Anticipation of somatosensory and motor events increases centro-parietal functional coupling: An EEG coherence study. *Clinical Neurophysiology, 117*, 1000-1008.

Babiloni, C., Infarinato, F., Marzano, N., Iacoboni, M., Dassù, F., Soricelli, A., . . . Del Percio, C. (2011). *intra-hemispheric functional coupling of alpha rhythms is related to golfer's prformance: a coherence EEG study*. Italy.

Babiloni.C, Del Percio, C., Iacoboni.M, Infarinato, F., Lizio, R., Marzano, N., . . . Eusebi, F. (2008). Golf putt outcomes are predictednby sensorimotor cerebral EEG rhythms. *The Journal of Physiology, 586*, 131-139.

Bahcivan, H., Hopkins, J., Zhang, N., Mirski, M., & Sherman, D. (2001). Cross-correlation analysis of epileptiform propagation using wavelets. *Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE* (pp. 1804-1807). IEEE.

Baumeister, J., Herwegen, H., Liesen, H., & Weiss, M. (2007). Complex Neuromonitoring in a Golf Putting Task. *Medicine and science in sports and exercise*.

Bernhard E. Boser, Isabelle M. Guyon, & Vladimir N. Vapnik. (1992). A Training Algorithm for Optimal Margin Classifiers. *ACM New York, NY, USA*, 144-152.

Chuanga, L.-Y., Huangb, C.-J., & Hunga, T.-M. (2013). The differences in frontal midline theta power between successful and unsuccessful basketball free throws of elite basketball players. *International Journal of Psychophysiology, 90*(3), 321-328.

Crews, D., Martin, J., Hart, E., & Piparo, A. (1991). *The Effectiveness of EEG Biofeedback, Relaxation, and Imagery Training on Golf Putting Performance.* Monterey, CA: North American Society for the Psychology of Sport and Physical Activity.

Davey, M., Victor, J., & Schiff, N. (2000). Power spectra and coherence in the EEG of a vegetative patient with severe asymmetric brain damage. *International Federation of Clinical Neurophysiology, 111*(2000), 1949-1954.

DJ, C., & DM., L. (1993). Electroencephalographic measures of attentional patterns prior to the golf putt. *medicine and science in sports and exercise, 25*(1), 116-26.

Drish, J. (1998). Obtaining Calibrated Probability Estimates from Support Vector Machine.

Hammond, D. C. (2007). Neurofeedback for the Enhancement of Athletic Performance. *The Journal of the American Board of Sport Psychology*.

Hermanto, B., Mengko, T., Indrayanto, A., & Prihatmanto, A. (2013). Brain signal reference concept using cross correlation based for brain computer interface. *Instrumentation,* *Communications, Information Technology, and Biomedical Engineering* (pp. 388 - 391). Bandung: IEEE.

Hillman, C. H., Apparies, R. J., Janelle, C. M., & Hatfield, B. D. (2000). An electrocortical comparison of executed and rejected shots in skilled marksmen. *biological psycology, 52*(1), 71-83.

Hsu, C.-W., Chang, C.-C., & Lin, a. C.-J. (2010). *A Practical Guide to Support Vector Classification.* National Taiwan University, Taipei 106, Taiwan.

Kirmizi-Alsana, E., Bayraktaroglu, Z., Gurvit, H., Keskin, Y. H., Emre, M., & Demiralp, T. (2006). Comparative analysis of event-related potentials during Go/NoGo and CPT: Decomposition of electrophysiological markers of response inhibition and sustained attention. *Brain Research*, 114-128.

Kousarrizi, M., Biomed. Eng. Dept., K. T., Ghanbari, A., Gharaviri, A., & Teshnehlab, M. (2009). Classification of Alcoholics and Non-Alcoholics via EEG Using SVM and Neural Networks. *Bioinformatics and Biomedical Engineering , 2009. ICBBE 2009. 3rd International Conference* (pp. 1-4). Beijing,China: IEEE.

Li, K., Zhang, X., & Du, Y. (2013). A SVM based classification of EEG for predicting the movement intent of human body. *Ubiquitous Robots and Ambient Intelligence (URAI), 2013 10th International Conference.* Jeju.

Lin, H.-T., Lin, C.-J., & Weng, R. C. (2007). A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 267-276.

Martijn Arns MS, Michiel Kleinnijenhuis MSc, Kamran Fallahpour , & Rien Breteler . (2008). Golf Performance Enhancement and Real-Life Neurofeedback Training Using Personalized Event-Locked EEG Profiles. *Journal of Neurotherapy, 11*(4), 11-18.

Muangjaroen, P., & Wongsawat, Y. (2012). Real-Time Index for Predicting of Successful Golf Putting Motion Using Multichannel EEG. *34th Annual Interntional Coference of the IEEE EMBS.* San Diego, California USA.

Niedermeyer, E., & Silva, F. H. (2012). *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields.* philadelphia,PA: Lippincott Williams & Wilkins; Fifth edition.

P. Terry, P. Mahoney. (2006). EEG correlates of golf performance: Training the brain to drain the putt. *Journal of Science and Medicine in Sport, 9,* 31.

Parvez, Z., & Paul, M. (2014). EEG Signal Classification using Frequency Band Analysis towards Epileptic Seizure Prediction. *IEEE International conference on Computer and Information Technology.* Bangladesh.

Paul, P., Leung, H., Peterson, D., Sejnowski, T., & Poizner, H. (2010). Detecting Neural Decision Patterns Using SVM-Based EEG Classification. *Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference.* Chengdu,China.

Platt, J. C. (1999). *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods.* Advances in large margin classifiers.

Rilk, A. J., Soekadar, S. R., Sauseng, P., & Plewnia, C. (2011). Alpha coherence predicts accuracy during a visuomotor tracking task. *Neuropsychologia*, 3704–3709.

Schlkopf, B., & Smola, A. J. (2002). Learning with kernels. In A. J. Bernhard Schlkopf, *Kernels* (pp. 32-48). London,England: The MIT Press.

Shaw, J. (1981). An introduction to the coherence function and its use in EEG signal analysis. *Journal of medical engineering & technology., 5*(6), 279-88.

Shen, K., Li, X., Ong, C., Shao, S., & Wilder-Smith, E. (2008). EEG-based mental fatigue measurement using multi-class support vector machines with confidence estimate. *Clinical neurophysiology, 119*(7), 524-33.

Tatum, W., Husain, A., Benbadis, S., & Kaplan, P. (2007). *Handbook of EEG Interpretation: 1st Edition.* New York: Demos Medical Publishing, LLC.

Thatcher, R. W., Biver, C. J., & M., D. (2004). *EEG and Brain Connectivity.* North,MA.

Tryer,L. (1988). *Biochemistry.* New York: W.H.

Vysata, O., Kukal, J., Prochazka, A., Pazdera, L., Simko, J., & Valis, M. (2014). Age-related changes in EEG coherence. *Polish Journal of Neurology and Neurosurgery, 48*(1), 35-8.

Appendix A Complete result tables

Table A: Binary classification for correlation coefficient

Features Random generator V5	**Optimal parameters (C, γ)**	**Cross-validation accuracy (%)**	**Training accuracy (%) confusion matrix**	**Testing accuracy (%) confusion matrix**
Cross Correlation (4-30Hz), seed [1]	C = 4 γ = 0.0089	57.1053	84.07 Confusion matrix =[178 18 43 44]	57.89 Confusion matrix = [63 34 46 47]
Cross Correlation (4-30Hz), seed [11]	C = 28.1 γ = 0.0356	58.1579	100 Confusion matrix =[196 0 0 187]	56.32 Confusion matrix = [48 49 34 59]
Cross Correlation (4-30Hz), seed [21]	C = 1.3543 γ = 0.1051	59.2105	99.22 Confusion matrix =[195 1 2 185]	51.05 Confusion matrix = [53 44 49 44]
Cross Correlation (4-30Hz), seed [31]	C = 6.7272 γ = 0.0156	57.3684	95.04 Confusion matrix =[194 2 17 170]	53.16 Confusion matrix = [62 35 54 39]
Cross Correlation (4-30Hz), seed [41]	C = 1 γ = 0.3536	56.8421	100 Confusion matrix =[196 0 0 187]	54.74 Confusion matrix = [88 9 77 16]
Cross Correlation (4-30Hz), seed [51]	C = 1.1388 γ = 0.0203	61.0526	84.60 Confusion matrix =[183 13 46 141]	50 Confusion matrix = [52 45 50 43]
Cross Correlation (4-30Hz), seed [61]	C = 4.1771 γ = 0.0482	58.9474	99.22 Confusion matrix =[195 1 2 185]	56.32 Confusion matrix =[55 42 41 52]
Cross Correlation (4-30Hz), seed [71]	C = 7.0250 γ = 0.0653	53.4211	99.48 Confusion matrix =[194 2 0 187]	58.95 Confusion matrix =[67 30 48 45]
Cross Correlation (4-30Hz), seed [81]	C = 61.2866 γ = 0.0010	58.4211	80.94 Confusion matrix =[173 23 50 137]	54.74 Confusion matrix =[55 42 44 49]
Cross Correlation (4-30Hz), seed [91]	C = 181.0193 γ =0.0010	57.6316	88.25 Confusion matrix =[183 13 32 155]	53.16 Confusion matrix =[57 40 49 44]
Cross Correlation (4-30Hz), seed [101]	C = 29.3441 γ =0.0012	56.3158	78.33 Confusion matrix =[169 27 56 131]	53.16 Confusion matrix =[66 31 58 35]

Table A (Cont.)				
Features Random generator V5	**Optimal parameters (C, γ)**	**Cross-validation accuracy (%)**	**Training accuracy (%) confusion matrix**	**Testing accuracy (%) confusion matrix**
Cross Correlation (4-30Hz), seed 111	C = 8.3542 γ = 0.0058	58.6842	84.86 Confusion matrix =[184 12 46 141]	55.79 Confusion matrix =[62 35 49 44]
Cross Correlation (4-30Hz), seed 121	C = 291.5299 γ = 0.0011	58.6842	94.52 Confusion matrix =[188 8 13 174]	48.42 Confusion matrix =[46 51 47 46]
Cross Correlation (4-30Hz), seed 131	C = 24.6754 γ = 0.0013	58.6842	79.37 Confusion matrix =[169 27 52 135]	53.68 Confusion matrix =[60 37 51 42]
Cross Correlation (4-30Hz), seed 141	C = 2.1810 γ = 0.0682	57.6316	98.69 Confusion matrix =[193 3 2 185]	48.95 Confusion matrix =[54 43 54 39]
Cross Correlation (4-30Hz), seed 151	C = 12.8839 γ = 0.0049	62.3684	87.47 Confusion matrix =[181 15 33 154]	51.58 Confusion matrix =[54 43 49 44]
Cross Correlation (4-30Hz), seed 161	C = 76.1093 γ = 0.0041	59.2105	97.91 Confusion matrix =[193 3 5 182]	49.47 Confusion matrix =[51 46 50 43]
Cross Correlation (4-30Hz), seed 171	C = 66.8335 γ = 0.0221	56.0526	100 Confusion matrix =[196 0 0 187]	56.32 Confusion matrix =[53 44 39 54]
Cross Correlation (4-30Hz), seed 181	C = 98.7015 γ = 0.0014	59.4737	88.25 Confusion matrix =[186 10 35 152]	53.68 Confusion matrix =[57 40 48 45]
Cross Correlation (4-30Hz), seed 191	C = 1 γ = 0.0313	57.3684	88.25 Confusion matrix =[190 6 39 148]	55.79 Confusion matrix =[63 34 50 43]
Cross Correlation (4-30Hz), seed 201	C = 122.5732 γ = 0.0013	56.0526	88.77 Confusion matrix =[181 15 28 159]	52.63 Confusion matrix =[48 49 41 52]

Table B: Binary classification for power spectrum density

Features Random generator V5	Optimal parameters (C, γ)	Cross-validation accuracy (%)	Training accuracy (%) confusion matrix	Testing accuracy (%) confusion matrix
PSD (4-30Hz), seed [1]	C = 8.192e+03 γ = 28.1	52.3684	86.16 Confusion matrix =[165 31 22 165]	53.68 Confusion matrix = [51 46 42 51]
PSD (4-30Hz), seed [11]	C = 1.579e+03 γ = 24.6754	56.3158	80.68 Confusion matrix =[165 31 43 144]	50.53 Confusion matrix = [54 43 51 42]
PSD (4-30Hz), seed [21]	C = 3.444e+03 γ = 12.3377	50.2632	77.55 Confusion matrix =[137 59 27 160]	47.89 Confusion matrix = [43 54 45 48]
PSD (4-30Hz), seed [31]	C = 3.2768e+04 γ = 32	48.4211	89.56 Confusion matrix =[181 15 25 162]	50 Confusion matrix = [57 40 55 38]
PSD (4-30Hz), seed [41]	C = 2.7554e+04 γ = 8	51.5789	83.29 Confusion matrix =[169 27 37 150]	51.58 Confusion matrix = [55 42 50 43]
PSD (4-30Hz), seed [51]	C = 1.166e+03 γ = 32	52.1053	79.63 Confusion matrix =[158 38 40 147]	43.16 Confusion matrix = [46 51 57 36]
PSD (4-30Hz), seed [61]	C = 4.871e+03 γ = 20.7494	48.9474	81.20 Confusion matrix =[168 28 44 143]	50 Confusion matrix =[56 41 54 39]
PSD (4-30Hz), seed [71]	C = 2.896e+03 γ = 30.6433	47.1053	80.16 Confusion matrix =[171 25 51 136]	50 Confusion matrix =[56 41 54 39]
PSD (4-30Hz), seed [81]	C = 6.049e+03 γ = 9.5137	52.8947	80.68 Confusion matrix =[151 45 29 158]	43.16 Confusion matrix =[45 52 56 37]
PSD (4-30Hz), seed [91]	C = 2.8774e+04 γ = 2.0885	41.5789	75.98 Confusion matrix =[157 39 53 134]	51.05 Confusion matrix =[60 37 56 37]
PSD (4-30Hz), seed [101]	C = 4.467e+03 γ = 32	49.4737	80.94 Confusion matrix =[165 31 42 145]	47.37 Confusion matrix =[47 50 50 43]

Table A (Cont.)				
Features Random generator V5	**Optimal parameters (C, γ)**	**Cross-validation accuracy (%)**	**Training accuracy (%) confusion matrix**	**Testing accuracy (%) confusion matrix**
PSD (4-30Hz), seed 111	C = 3.2768e+04 γ = 5.4170	50	83.03 Confusion matrix =[170 26 39 148]	44.21 Confusion matrix =[47 50 56 37]
PSD (4-30Hz), seed 121	C = 2.543e+03 γ = 26.9087	46.3158	75.72 Confusion matrix =[140 56 37 150]	49.47 Confusion matrix =[36 61 35 58]
PSD (4-30Hz), seed 131	C = 2.543e+03 γ = 12.8839	55.2632	79.37 Confusion matrix =[142 54 25 162]	41.05 Confusion matrix =[32 65 47 46]
PSD (4-30Hz), seed 141	C = 2.1247e+04 γ = 16	50	86.42 Confusion matrix =[171 25 27 160]	46.84 Confusion matrix =[43 54 47 46]
PSD (4-30Hz), seed 151	C = 2.5268e+04 γ = 20.7494	51.3158	89.06 Confusion matrix =[175 21 21 166]	49.47 Confusion matrix =[53 44 52 41]
PSD (4-30Hz), seed 161	C = 1.117e+03 γ = 32	54.4737	78.33 Confusion matrix =[174 22 61 126]	47.89 Confusion matrix =[60 37 62 31]
PSD (4-30Hz), seed 171	C = 2.7554e+04 γ = 16.7048	51.8421	88.25 Confusion matrix =[181 15 30 157]	46.84 Confusion matrix =[51 46 55 38]
PSD (4-30Hz), seed 181	C = 8.192e+03 γ = 32	53.6842	85.12 Confusion matrix =[169 27 30 157]	48.95 Confusion matrix =[47 50 47 46]
PSD (4-30Hz), seed 191	C = 2.1247e+03 γ = 4.3620	54.7368	81.72 Confusion matrix =[165 31 39 148]	50.53 Confusion matrix =[48 49 45 48]
PSD (4-30Hz), seed 201	C = 2.774e+03 γ = 16.7048	50.2632	77.28 Confusion matrix =[178 18 69 118]	51.58 Confusion matrix =[55 42 50 43]

Table C: Binary classification for coherence

Features Random generator V5	Weight (w1 w2) Optimal parameters (C, γ)	Cross-validation accuracy (%)	Training accuracy (%) confusion matrix	Testing accuracy (%) confusion matrix
Coherence (4-30Hz), seed [1]	C = 3.0048e+04 γ = 0.001	60	86.42 Confusion matrix =[171 25 27 160]	54.74 Confusion matrix = [52 45 41 52]
coherence (4-30Hz), seed [11]	C = 29 γ = 1.4768	61.0526	100 Confusion matrix =[196 0 0 187]	52.63 Confusion matrix = [44 53 37 56]
coherence (4-30Hz), seed [21]	C = 1 γ = 1.7563	60.5263	96.87 Confusion matrix =[194 2 10 177]	54.21 Confusion matrix = [32 65 22 71]
Coherence (4-30Hz), seed [31]	C = 1.117e+03 γ = 0.0033	59.4737	79.63 Confusion matrix =[166 30 48 139]	47.89 Confusion matrix = [50 47 52 41]
coherence (4-30Hz), seed [41]	C = 8 γ = 3.3636	60.2632	99.74 Confusion matrix =[195 1 0 187]	50.53 Confusion matrix = [33 64 30 63]
Coherence (4-30Hz), seed [51]	C = 1722 γ = 0.0022	58.1579	79.90 Confusion matrix =[163 33 44 143]	55.79 Confusion matrix = [63 34 50 43]
Coherence (4-30Hz), seed [61]	C = 1.5024e+04 γ = 0.0097	56.8421	99.74 Confusion matrix =[195 1 0 187]	50 Confusion matrix =[51 46 49 44]
Coherence (4-30Hz), seed [71]	C = 1 γ = 1.1388	58.6842	93.21 Confusion matrix =[184 12 14 173]	52.63 Confusion matrix =[36 61 29 64]
Coherence (4-30Hz), seed [81]	C = 2 γ = 1.834	55.2632	99.22 Confusion matrix =[194 2 1 186]	62.11 Confusion matrix =[56 41 31 62]
Coherence (4-30Hz), seed [91]	C = 4 γ = 2.7085	57.8947	99.74 Confusion matrix =[196 0 1 186]	52.63 Confusion matrix =[34 63 27 66]
Coherence (4-30Hz), seed [101]	C = 609 γ = 0.0072	57.8947	82.77 Confusion matrix =[167 29 37 150]	56.32 Confusion matrix =[59 38 45 48]
Coherence (4-30Hz), seed [111]	C = 2 γ = 1.7563	56.5789	99.48 Confusion matrix =[195 1 1 186]	54.74 Confusion matrix =[43 54 32 61]

Table C (Cont.)

Features Random generator V5	Weight (w1 w2) Optimal parameters (C, γ)	Cross-validation accuracy (%)	Training accuracy (%) confusion matrix	Testing accuracy (%) confusion matrix
Coherence (4-30Hz), seed [121]	C = 59 γ = 0.5	56.5789	99.74 Confusion matrix =[195 1 0 187]	50.53 Confusion matrix =[50 47 47 46]
Coherence (4-30Hz), seed [131]	C = 1 γ = 1.2968	56.5789	96.61 Confusion matrix =[192 4 9 178]	56.32 Confusion matrix =[51 46 37 56]
Coherence (4-30Hz), seed [141]	C = 9 γ = 0.5453	58.9474	99.74 Confusion matrix =[195 1 0 187]	51.58 Confusion matrix =[47 50 42 51]
Coherence (4-30Hz), seed [151]	C = 1.5024 γ = 0.0014	55.7895	84.33 Confusion matrix =[167 29 31 156]	53.16 Confusion matrix =[54 43 46 47]
Coherence (4-30Hz), seed [161]	C = 18 γ = 9.766e-04;	59.7368	89.56 Confusion matrix =[182 14 26 161]	51.05 Confusion matrix =[56 41 52 41]
Coherence (4-30Hz), seed [171]	C = 52 γ = 1.4142	53.9474	99.74 Confusion matrix =[196 0 1 186]	56.84 Confusion matrix =[38 59 23 70]
Coherence (4-30Hz), seed [181]	C = 431 γ = 0.0313	57.6316	95.82 Confusion matrix =[187 9 7 180]	49.47 Confusion matrix =[41 56 40 53]
Coherence (4-30Hz), seed [191]	C = 1 γ = 1.1892	61.3158	95.56 Confusion matrix =[189 7 10 177]	52.63 Confusion matrix =[53 44 46 47]
Coherence (4-30Hz), seed [201]	C = 1.328e+03 γ = 0.0024	58.9474	79.11 Confusion matrix =[163 33 47 140]	57.89 Confusion matrix =[59 38 42 51]

Table D: Binary classification for average coherence

Features Random generator V5	Optimal parameters (C, γ)	Cross-validation accuracy (%)	Training accuracy (%) confusion matrix	Testing accuracy (%) confusion matrix
Average coherence seed [1]	C = 6.09e+02 γ = 1.0443	57.8947	99.74 Confusion matrix =[195 1 0 187]	54.21 Confusion matrix = [55 42 45 48]
Average coherence seed [11]	C = 6 γ = 32	56.8421	99.74 Confusion matrix =[195 1 0 187]	50.53 Confusion matrix = [30 67 27 66]
Average coherence seed [21]	C = 2 γ = 13.4543	60.7895	97.13 Confusion matrix =[194 2 9 178]	57.89 Confusion matrix = [54 43 37 56]
Average Coherence seed [31]	C = 7 γ = 1.1388	59.7368	79.90 Confusion matrix =[174 22 55 132]	52.11 Confusion matrix = [53 44 47 46]
Average Coherence seed [41]	C = 5 γ = 3.0844	60.2632	89.82 Confusion matrix =[182 14 25 162]	48.42 Confusion matrix = [50 47 51 42]
Average Coherence seed [51]	C = 15 γ = 0.8781	60.7895	83.03 Confusion matrix =[174 22 43 144]	52.11 Confusion matrix = [60 37 54 39]
Average Coherence seed [61]	C = 636 γ = 0.0743	60.7895	79.90 Confusion matrix =[165 31 46 141]	52.63 Confusion matrix =[52 45 45 48]
Average Coherence seed [71]	C = 70 γ = 12.3377	59.7368	78.85 Confusion matrix =[147 49 32 155]	52.63 Confusion matrix =[33 64 26 67]
Average Coherence seed [81]	C = 2.5268e+04 γ = 0.015	56.0526	80.68 Confusion matrix =[160 36 38 149]	60 Confusion matrix =[62 35 41 52]
Average Coherence seed [91]	C = 159 γ = 0.3242	63.6842	85.38 Confusion matrix =[177 19 37 150]	51.05 Confusion matrix =[49 48 45 48]
Average Coherence seed [101]	C = 4 γ = 8.7241	57.6316	97.65 Confusion matrix =[193 3 6 181]	52.11 Confusion matrix =[50 47 44 49]

Table D (Cont.)				
Features Random generator V5	**Optimal parameters (C, γ)**	**Cross-validation accuracy (%)**	**Training accuracy (%) confusion matrix**	**Testing accuracy (%) confusion matrix**
Average Coherence seed [111]	C = 2.543e+03; γ = 0.1928	54.7368	94.26 Confusion matrix =[187 9 13 174]	56.32 Confusion matrix =[55 42 41 52]
Average Coherence seed [121]	C = 25 γ = 13.4543	58.4211	99.74 Confusion matrix =[195 1 0 187]	54.21 Confusion matrix =[50 47 40 53]
Average Coherence seed [131]	C = 2.0347e+04; γ = 0.1621	58.4211	98.69 Confusion matrix =[192 4 1 186]	52.63 Confusion matrix =[53 44 46 47]
Average Coherence seed [141]	C = 1 γ = 5.1874	59.7368	83.55 Confusion matrix =[174 22 41 146]	58.95 Confusion matrix =[63 34 44 49]
Average Coherence seed [151]	C = 70 γ = 0.4585	58.1579	80.68 Confusion matrix =[161 35 39 148]	60 Confusion matrix =[54 43 33 60]
Average Coherence seed [161]	C = 13 γ = 0.8409	62.1053	82.77 Confusion matrix =[170 26 40 147]	53.16 Confusion matrix =[55 42 47 46]
Average Coherence seed [171]	C = 512 γ = 0.5	55.2632	94.26 Confusion matrix =[183 13 9 178]	50.53 Confusion matrix =[50 47 47 46]
Average Coherence seed [181]	C = 2.7554e+04; γ = 0.1621	59.2105	99.22 Confusion matrix =[193 3 0 187]	50 Confusion matrix =[41 56 39 54]
Average Coherence seed [191]	C = 279 γ = 0.4788	61.3158	90.86 Confusion matrix =[177 19 6 171]	53.68 Confusion matrix =[57 40 48 45]
Average Coherence seed [201]	C = 6 γ = 11.3137	58.9474	99.74 Confusion matrix =[195 1 0 187]	50.53 Confusion matrix =[48 49 45 48]

Appendix B   Source code

```
clc;
close all;
clear all;
%% input data
powerdata_all = csvread('all573_time128_coherence_4_30.csv',0,0);
decision_data_all = xlsread('all573_performance30withtext_official.xlsx','D2:E574');
%% important condition
percentage_bound = 80 ;
%% scaling the original data
x_all = (powerdata_all - repmat(min(powerdata_all,[],1),size(powerdata_all,1),1))*...
        spdiags(1./(max(powerdata_all,[],1)-
min(powerdata_all,[],1))',0,size(powerdata_all,2),size(powerdata_all,2));

%% pre-label by Qualities and centmeters
for i = 1:573
if  decision_data_all(i,1) >= 8 && decision_data_all(i,2)<=30;
   decision_data(i) = 1; %good
elseif decision_data_all(i,1) <= 7 && decision_data_all(i,2)>=50;
   decision_data(i) = 3; %bad
else
   decision_data(i) = 2; %ok
end
end
y_all = decision_data;

%% seperate training data and testing data
seed              = 81 ; % seed generate different random sequency
all_c_train_matrix =[];        % for recording all training confusion matrix
all_c_test_matrix  =[];        % for recording all training confusion matrix
for h = 1:length(seed)

rand('state',seed(h)); randn('state',seed(h));
%1/3 is test data and 2/3 is training data, the ratio is fixed
finaltest_index = [randsample(find(y_all==1),floor(length(find(y_all==1))/3))...
                randsample(find(y_all==2),floor(length(find(y_all==2))/3))...
                randsample(find(y_all==3),floor(length(find(y_all==3))/3))];
x_finaltest    = x_all(finaltest_index,:);
y_finaltest    = y_all(finaltest_index)';


Train_index      = setdiff(1:length(y_all),finaltest_index);
x_Train_all1     = x_all(Train_index,:);
y_Train_all1     = y_all(Train_index)';
```

```
%make sure the first group that input the structure is labeled 1
[y_Train_all,right_order] = sort(y_Train_all1);
x_Train_all              = x_Train_all1(right_order,:);

%% ratio in five subsets is fixed and the examples # is fixed
index_s    = find(y_Train_all==1);
index_a    = find(y_Train_all==2);
index_f    = find(y_Train_all==3);

number_of_success = length(index_s);
number_of_average = length(index_a);
number_of_failure = length(index_f);

alpha=1-number_of_success/(number_of_success+number_of_average+number_of_failure);
xValidationFolds = 5;

% make sure the length in each fold is the same
cloumn1      =  floor(length(index_s)/xValidationFolds);
cloumn2      =  floor(length(index_a)/xValidationFolds);
cloumn3      =  floor(length(index_f)/xValidationFolds);

rand('state',1); randn('state',1);
index_ss       =  reshape(index_s(randperm(5*cloumn1)),5,cloumn1);
index_aa       =  reshape(index_a(randperm(5*cloumn2)),5,cloumn2);
index_ff       =  reshape(index_f(randperm(5*cloumn3)),5,cloumn3);

%combine 2 and 3 together as 2
y_finaltest(find(y_finaltest==3))=2;
y_Train_all(find(y_Train_all==3))=2;

%% start search
for search = 1:5
% set up c and sigma for each searching
if search == 1
% original C    step 1
step1 =1 ;
min_c = -5; max_c =15;
C_array = 2.^(min_c:step1:max_c);
% original Sigma step 1
min_sigma = -10; max_sigma = 5;
Sigma_array = 2.^(min_sigma:step1:max_sigma);
G_length = length(Sigma_array);
else
step1 = step1/2;
% new c
C_range = max(min_c, log2(best_C)-step1*10):step1:min(log2(best_C)+step1*10,max_c);
```

```
C_array = 2.^(C_range);
% new sigma
Sigma_range = max(min_sigma, log2(best_sigma)-
step1*10):step1:min(log2(best_sigma)+step1*10,max_sigma);
Sigma_array=2.^(Sigma_range);
end
C_length = length(C_array);
G_length=length(Sigma_array);
%%
% 3 important value
train_correct_rate = zeros(C_length, G_length);
test_correct_rate  = zeros(C_length, G_length);
optimal_weight     = zeros(C_length, G_length);

%% training start
for k=1:C_length
    for j=1:G_length
        for fold=1:xValidationFolds
            testIndex    = [index_ss(fold,:),index_aa(fold,:),index_ff(fold,:)];
            trainIndex   = setdiff(1:length(y_Train_all),testIndex);

            x_train      = x_Train_all(trainIndex,:) ;
            y_train      = y_Train_all(trainIndex);

            x_test       = x_Train_all(testIndex,:);
            y_test       = y_Train_all(testIndex);

            Parameters   = ['-c ' num2str(C_array(k)) ' -g ' num2str(Sigma_array(j)) ' -b 0'];
            model        = libsvmtrain(y_train,x_train,Parameters);


           [label_train1,~,group_train] = libsvmpredict(y_train, x_train, model,'-b 0'); %revised by
baohua %[~,~,group_train] = libsvmpredict(y_train, x_train, model);
           [label_test1,~,group]        = libsvmpredict(y_test, x_test, model, '-b 0');  %revised by
baohua %[~,~,group]       = libsvmpredict(y_test, x_test, model);

           %%
            % training accuracy
            train_correct_rate(k,j) =
train_correct_rate(k,j)+100*sum(label_train1==y_train)/size(x_train,1);
            % test accuracy
            test_correct_rate (k,j) =
test_correct_rate(k,j)+100*sum(label_test1==y_test)/size(x_test,1);

            Pro_test_1              =
length(intersect(find(y_test==1),find(label_test1~=1)))/length(find(y_test==1));
```

```
            if isnan(Pro_test_1);
                Pro_test_1 = 0;
            end

            Pro_test_2              =
length(intersect(find(y_test==2),find(label_test1~=2)))/length(find(y_test==2));
            if isnan(Pro_test_2);
                Pro_test_2 = 0;
            end

            optimal_weight(k, j)    = optimal_weight(k, j)+alpha*Pro_test_1+(1-alpha)*Pro_test_2;
        end % end fold
        train_correct_rate(k,j)     = train_correct_rate(k,j)/xValidationFolds;
        test_correct_rate(k,j)      = test_correct_rate(k,j)/xValidationFolds;
        optimal_weight(k, j)        = optimal_weight(k,j)/xValidationFolds;  % as high as possible
        %%%%%%%%%%%%%%%
        if train_correct_rate(k,j)  < percentage_bound
           optimal_weight(k, j)     = Inf;
        end
        %%%%%%%%%%%%%%%
    end % end j
end %end k

[best_rate1,index1] = min(optimal_weight);
[best_rate2,index2] = min(best_rate1);
best_C              = C_array(index1(1, index2));
best_sigma          = Sigma_array(index2);
cross_validation    =test_correct_rate(index1(1, index2), index2);
end

Final_Parameters          = ['-c ' num2str(best_C) ' -g ' num2str(best_sigma) ' -b 0'];
final_model               = libsvmtrain(y_Train_all,x_Train_all,Final_Parameters);

[label_train,~,p_train]   = libsvmpredict(double(y_Train_all),x_Train_all, final_model,'-b 0');
[label_test,~,p_test]     = libsvmpredict(double(y_finaltest),x_finaltest, final_model,'-b 0');

train_acc                 = sum(label_train == y_Train_all) ./ numel(y_Train_all)
test_acc                  = sum(label_test == y_finaltest) ./ numel(y_finaltest)
C_train                   = confusionmat(y_Train_all,label_train)
C_test                    = confusionmat(y_finaltest,label_test)


allc(h)  = best_C
allsigma(h) = best_sigma
bestrate(h) = best_rate2
cross_validation_all(h) = cross_validation
```

```
all_train_acc(h) = train_acc
all_test_acc(h) = test_acc
all_c_train_matrix = [all_c_train_matrix, C_train]
all_c_test_matrix  = [all_c_test_matrix, C_test]
end
savefile = 'Thesis_psd_2class_lin'
save(savefile,'allc','allsigma','bestrate','cross_validation','all_train_acc','all_test_acc',...
    'all_c_train_matrix','all_c_test_matrix')
```