Dissertation for Doctor of Philosophy

Robust Sparse Representation based Classification Scheme for Non-stationary EEG Signal Classification

Younghak Shin

School of Information and Communications

Gwangju Institute of Science and Technology

2016

박 사 학 위 논 문

비정상성 뇌파 신호 분류를 위한 강력한 희소화 표현 기반 분류 방법

신 영 학

정 보 통 신 공 학 부

광 주 과 학 기 술 원

2 0 1 6

To my parents and my lovely wife, Hwajin,

for their devoted support and love

Ph.D/ IC 20112099 Younghak Shin. Robust Sparse Representation based Classification Scheme for Non-stationary EEG Signal Classification. School of Information and Communications. 2016. 83p. Advisor: Prof. Heung-No Lee.

Abstract

Brain-Computer Interface (BCI) systems provide an alternative communication and control channel between the human brain and external devices, such as computer programs or prosthetic hands. In the BCI systems, electroencephalography (EEG) signals are widely used for recording of user's intention or imagination generated by electrical activity along the scalp. However, scalp-recorded EEG signals have inherent non-stationary characteristics; thus, the classification performance is deteriorated by changing the background activity of the EEG during the BCI experiment. Therefore, powerful signal processing methods are needed for reliable BCI performance.

In this thesis, first, we propose a new sparse representation-based classification (SRC) scheme for motor imagery (MI)-based BCI applications. The proposed SRC method utilizes the frequency band power and CSP algorithm to extract MI features for classification. In SRC, the design of a good dictionary matrix is critical. We provide a detailed design procedure for constructing the dictionary matrix for SRC scheme. Second, we analyze noise robustness of the SRC method to evaluate the capability of the SRC for non-stationary EEG signal classification. Using the noisy test signals generated by Gaussian noise and background noise, we compare the classification performance of the

SRC and support vector machine (SVM). Furthermore, we analyze the unique classification mechanism of the SRC. Third, to overcome non-stationary effects of EEG signals, we propose simple adaptive sparse representation based classification (SRC) schemes. Supervised and unsupervised dictionary update techniques for new test data and a dictionary modification method by using the incoherence measure of the training data are investigated. The proposed methods are very simple and additional computation for the re-training of the classifier is not needed. The proposed adaptive SRC schemes are evaluated using two BCI experimental datasets. The proposed methods are assessed by comparing classification results with the conventional SRC and other LDA and SVM adaptive classification methods.

Ph.D/ IC 20112099 신 영 학. 비정상성 뇌파 신호 분류를 위한 강력한 희소화 표현 기반 분류 방법. 정보통신공학부. 2016. 83p. 지도교수: 이 흥 노.

국문요약

뇌-컴퓨터 인터페이스 시스템은 사람의 뇌와 컴퓨터 프로그램 이나 보철 손과 같은 외부 장치들 사이의 대안적인 통신과 조정 채널을 제공한다. 뇌-컴퓨터 인터페이스 시스템에서 뇌파 신호는 사용자의 의도나 상상을 반영하는 전기적인 신호를 두피에서 레코딩하기 위해 가장 널리 사용된다. 그러나, 두피에서 획득한 뇌파 신호는 고유한 비정상성 특성을 갖는다. 즉, 뇌-컴퓨터 인터페이스 실험 동안 뇌파의 배경 활동 변화로 인해 분류 성능이 저하된다. 그러므로 신뢰할 수 있는 성능을 위해서 강력한 신호처리 방법들이 요구된다.

본 학위논문에서 첫 번째로 우리는 움직임 상상 기반의 뇌-컴퓨터 인터페이스 시스템을 위한 새로운 희소화 표현 기반 분류방법 (SRC)을 제안하였다. 제안된 희소화 표현 기반 분류법에서 주파수 밴드파워와 공통공간필터링 (CSP) 알고리즘을 움직임 상상신호의 특징추출을 위해 사용한다. 우리는 희소화 표현 기반 분류법에서 분류를 위해 가장 중요하게 사용되는 사전 (dictionary)을 만들기 위한 자세한 디자인 절차를 설명한다. 두 번째로, 우리는 희소화 표현 기반 분류법의 비정상성 뇌파 신호 분류 성능을 평가하기 위해 잡음 강건성을 분석한다. 가우시안 잡음과 배경잡음을 이용하여 생성된 잡음테스트 신호를 이용하여, 희소화 표현 기반 분류법과 서포트 벡터 머신(SVM) 분류법의 분류 성

능을 비교한다. 추가적으로, 우리는 희소화 표현 기반 분류법의 독특한 분류 메커니즘을 분석한다. 세 번째로, 뇌파 신호의 비정상성 효과를 극복하기 위하여 우리는 간단한 적응형 희소화 표현 기반 분류법들을 제안한다. 새로운 테스트 신호에 대한 지도 및 비지도 사전 업데이트 방법과 훈련데이터의 일관성(coherence) 측정을 통한 사전변경방법에 대해 조사한다. 제안된 적응형 분류 방법은 매우 간단하며 재-훈련 (re-training)을 위한 추가적인 계산이 필요하지 않다. 제안된 방법은 두 종류의 실험 데이터를 이용하여 기존의 희소화 표현 기반 분류법과 다른 적응형 선형분류 방법들 (LDA and SVM)과 비교를 통하여 성능을 평가한다.

Contents

List of Tables

List of Figures

1. Introduction

1.1 EEG based Brain-Computer Interface

Brain-computer interface (BCI) systems provide an alternative communication and control channel between human brain and external devices such as computer program and prosthetic device without any normal muscle movements [1]. Due to the convenient usability and high temporal resolution compared to other brain imaging schemes such as functional magnetic resonance imaging (fMRI) and magnetoencephalogram (MEG), research of noninvasive electroencephalogram (EEG) based brain–computer interface (BCI) systems is continuously progressed over the past several decades [1][2][3].

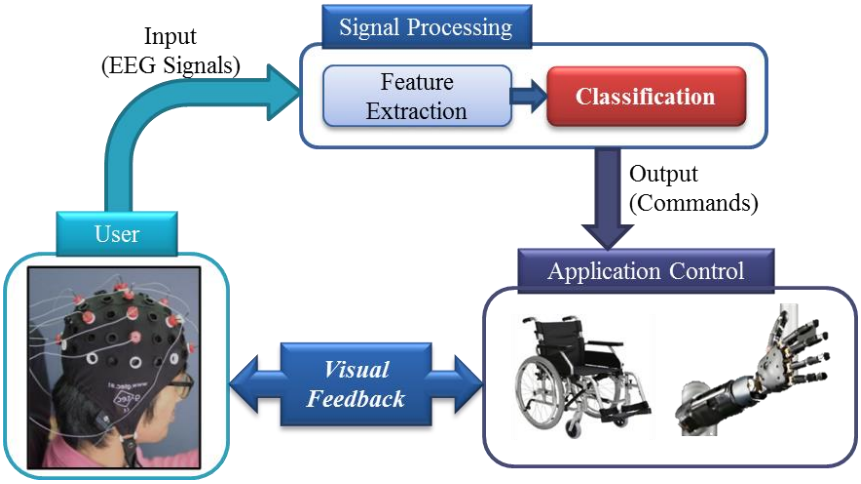


Figure 1.1 EEG based Brain-Computer Interface

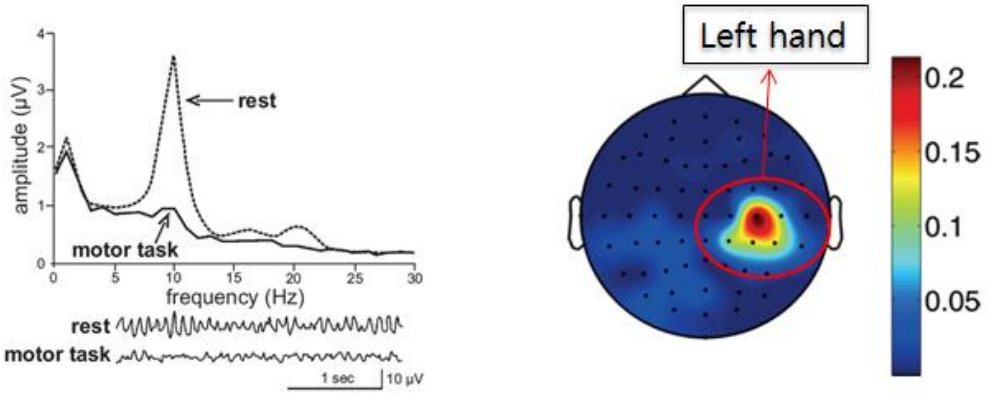
In the EEG based BCI systems, using scalp recorded EEG signals as an input of the system, essential signal processing methods are performed to translate user's intention into a computer command, which can then be used to control external devices. The signal processing steps in BCI can be categorized as preprocessing, feature extraction, and classification. In the preprocessing step, the artifact detection and rejection are conducted. The purpose of feature extraction is to make a

meaningful low-dimensional data, i.e., a feature vector, from the original high-dimensional data. This feature vector should be distinguishable for different classes. Typically, the feature extraction is performed using a dimensionality reduction method. The principal component analysis (PCA), independent component analysis (ICA), and common spatial pattern (CSP) are popular methods for dimensionality reduction in the EEG based BCI systems [4][5]. Another important signal processing step is classification. In the BCI systems, the purpose of classification is to translate the extracted feature into a computer command. Typically, this translation is done using the classification algorithms, which are adopted from pattern recognition area. Frequently used classification methods in the EEG based BCI systems are linear classifiers such as linear discriminant analysis (LDA) and support vector machine (SVM) [6].

1.2 Motor imagery Brain-Computer Interface

In this study, we focused on motor imagery (MI)-based BCI. EEG based MI is one of widely used BCI schemes [2][3]. MI scheme use sensorimotor rhythms (SMRs), such as the mu(8-12Hz) and/or beta(15~30Hz) rhythms; these rhythms can be recorded on the scalp over the sensorimotor cortex area when subject imagine their limb movements. A widely used feature in MI based BCIs is event-related desynchronization (ERD) [7]. Thus, a significant decrease in the power level of SMRs can be observed on the contralateral hemisphere during the unilateral imagination of hand movements. Figure 1.2 shows the SMR based MI features in terms of the spectral and spatial difference [1]. For example, when subjects imagine their left hand movement, a distinct feature, i.e., amplitude attenuation of Mu rhythm, appears over the contralateral hand area at the sensorimotor cortex. The

different patterns present in EEG signals are detected and used for BCI control. In this study, we use EEG signals from the MI experiment and analyze them for BCI signal pattern classification.



(a) (b)

Figure 1.2 Mu rhythm based BCI features. (a) Spectral difference between motor task and rest in the Mu rhythm (8~15Hz). (b) Spatial difference between motor task (Left hand motor imagery) and rest (Figure taken from [1]).

1.3 Sparse Representation based Classification

Recently, with much progress of L1 minimization technique in compressive sensing (CS) field [8][9], sparse representation has received a lot of attention in signal processing and pattern recognition fields. The problem of sparse representation is to find the most compact representation of a signal in terms of linear combination of columns in an over-complete dictionary. Thus, given a signal $\mathbf{y} \in \mathbb{R}^{m\times 1}$ and over-complete dictionary $\mathbf{A} \in \mathbb{R}^{m\times n}$ where $m \ll n$, sparse representation aims to find sparse coefficient $\mathbf{x} \in \mathbb{R}^{n\times 1}$ via the so-called sparsification step, i.e., $\min \|\mathbf{x}\|_1$ subject to $\mathbf{y} = \mathbf{A}\mathbf{x}$.

Especially, sparse representation based classification (SRC) has shown an increased interest. SRC framework is first introduced by Huang et al [10]. The basic idea of SRC is to parsimoniously

represent a test signal **y** via the sparsification step, i.e., **y** = **Ax**, where **A** is an over-complete dictionary whose columns are a collection of training signals. This sparsification step leads to the representation of the test signal **y** with the training signals from the same class predominantly. The L1 minimization algorithm is employed to perform the sparse representation of the test signal with a given set of training signals. The robust classification performance of the SRC framework has been shown in various applications such as face recognition [11][12], digit classification [10], and speech recognition [13]. Particularly, in [11], Yang et al. presented that SRC obtains robust face recognition performance for occlusion and corruption on facial images.

1.4 Outline of this Thesis

This dissertation consists of three research topics. In the first part we propose a sparse representation based classification (SRC) scheme for EEG based motor imagery BCI in Section 2 which was published in [14].

[14] **Y. Shin**, S. Lee, J. Lee and H.-N. Lee, "Sparse representation-based classification scheme for motor imagery-based brain-computer interface systems", *Journal of Neural Engineering*, no. 9 056002, 2002.

In the second part of this dissertation, we evaluate noise robustness of SRC for non-stationary EEG signal classification in Section 3, for more details, see [15].

[15] **Y. Shin**, S. Lee, M. Ahn, H. Cho, S. C. Jun and H.-N. Lee, "Noise Robustness Analysis of Sparse Representation based Classification Method for Non-stationary EEG Signal Classification", *Biomedical Signal Processing and Control* vol. 21, pp. 8-18, 2015.

The last part of this dissertation considers the adaptive SRC schemes for online BCI applications in Section 4, for more details, see [16].

[16] **Y. Shin**, S. Lee, M. Ahn, H. Cho, S. C. Jun and H.-N. Lee, "Simple Adaptive Sparse Representation based Classification Schemes for EEG based Brain-Computer Interface Applications", *Computers in Biology and Medicine*, vol. 66, pp. 29-38, 2015.

2. Sparse Representation based classification method for motor imagery BCI

2.1 Motivation

The EEG signals acquired from scalp electrodes are usually very noisy and show a non-stationary characteristic [17]. They contain signals from non-interesting physiological activities (e.g. electromyograms (EMGs)), the sensor noise present in any electrical system and environmental noise (e.g. power lines).

In particular, in the case of motor imagery based BCI, which uses induced EEG signals while the subject imagines limb movements, the instability of imagery task, non-stationarity of signals, and lack of concentration are among main obstacles to effectively process the EEG signals.

In addition, it is difficult to collect a large set of training samples because of the subject's fatigue. The raw EEG signals are associated with high dimension owing to the large number of EEG channels; hence, it is difficult to collect volume of data samples that are large enough for good training. Therefore, the use of powerful signal processing and classification techniques plays a critical role.

In this thesis, first we introduce a SRC method to EEG-based MI BCI applications. We use a band power approach that involves extracting the power information from the signal for the SMRs [18]. In SRC, the design of a good dictionary matrix is important, or the performance will be poor. We provide a detailed design procedure for constructing the dictionary matrix for EEG classification. To

maximize the benefit of sparse representation, we propose to use CSP filtering and preprocess the raw training signals to construct the columns of the dictionary.

2.2 Experimental dataset

In this study, we used two different datasets. The first was the INFONET dataset obtained from our own MI-based BCI experiment. The other was the Berlin dataset downloaded from the website of BCI competition III (dataset IVa) [19]. The main difference between the two datasets was that they used different numbers of EEG channels and had different sizes (i.e. numbers of total trials). The Berlin dataset contained more trials, i.e. 80 for the INFONET dataset and 140 for the Berlin dataset per class. The number of EEG electrodes used to collect data was also different, i.e. 16 for the INFONET dataset and 118 for the Berlin dataset.

INFONET dataset consisted of five different datasets obtained from five healthy subjects (five males, average age = 22 and SD = 6.85). They were all novice subjects in BCI experiments. There were two classes, i.e. the left- and right-hand motor imaginary movements. In this experiment, we used 16 EEG channels. The EEG signals were recorded from active electrodes in a cap (with the earlobe used as the reference) based on the international 10/20 standard. In our experiment, we used a g.EEGcap and g.ACTIVE electrodes made by G. Tec Inc. and a PZ3 amplifier from Tucker-Davis Technologies. We used a sampling rate of 256 samples s−1 with a band-pass filter of 1–100 Hz and a notch filter of 60 Hz.

In our BCI experiments, the subjects were seated in a comfortable chair and asked to watch a monitor screen. Figure 2.1 shows the time procedure for one trial. At the beginning of each run, a 'Left Hand' or 'Right Hand' letter instruction randomly appeared for 4 s at the center of the screen. Then, subjects imagined a left- or right-hand movement after the instruction was given, i.e. repeated fist clenching. This was followed by a rest period of 3 s. One run consisted of 40 trials, i.e. 20 left- and 20 right-hand trials. With all subjects, we conducted six runs that consisted of two runs with real movements and four runs with imaginary movements. We used only the imaginary data trials for further signal processing. To suppress electrooculogram (EOG) artifacts, the subjects were instructed not to blink or move their eyes during the instruction period. During the rest period, they could blink freely but were not allowed to move their body.

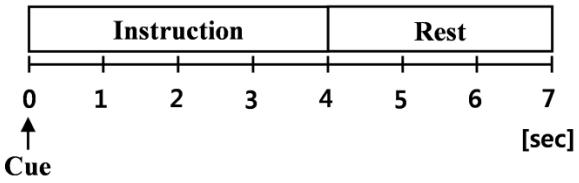


Figure 2.1 Experimental time procedure for INFONET dataset

Berlin dataset was produced in the BCI competition and is widely used in the BCI field for the analysis of EEG signal processing. It contains five datasets recorded from five different healthy subjects (*aa, al, av, aw, and ay*). The subjects followed the same procedure as the BCI experiment with three classes, i.e. left-hand, right-hand and right-foot MI movements. However, only the data corresponding to the right hand (R) and right foot (F) were used for analysis. These datasets only contain data from the four initial sessions without feedback. The data were recorded using BrainAmp

amplifiers and a 128-channel Ag/AgCl electrode cap from ECI. 118 EEG channels were measured at the positions of the extended international 10/20 system. The exact electrode positions are provided in the data file. The signals were band-pass filtered between 0.05 and 200 Hz and then digitized at 1000 Hz. The signals were downsampled to 100 Hz for offline analysis by the Berlin research team.

Figure 2.2 shows a timed trial procedure for the Berlin dataset. The subjects were seated in a comfortable chair with their arms resting on armrests. Visual cues were provided for 3.5 s that indicated the appropriate motor imagery the subject should perform, i.e. left hand, right hand, or right foot. The presentation of target cues alternated with periods of random length, i.e. 1.75–2.25 s, in which the subject could relax. Each of the five datasets consists of 140 trials for each class.

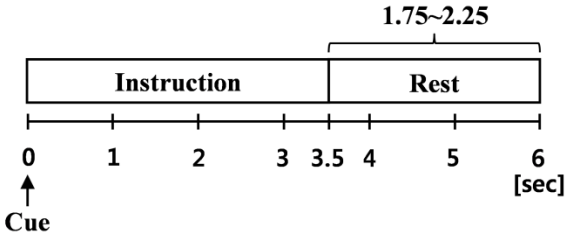


Figure 2.2 Experimental time procedure for Berlin dataset

2.3 Preprocessing and CSP filtering

Figure 2.3 shows the entire signal processing procedure, including the preprocessing, feature extraction and classification steps. Using the motor imagery dataset obtained from each subject, we perform the data preprocessing. Before preprocessing, raw EEG signals are segmented. After an instruction (left or right hand) appears on the screen, the time samples from 1 to 2s were collected for all trial data. We then apply the band pass filter to the trial data to eliminate the frequencies that are

not related to motor imagery signals. In this study, sensorimotor rhythm, 8–15 Hz, is used for cut off frequencies. We use fourth order Butterworth filter for band pass filtering. For fair comparison of the classification performance, we fixed the time and frequency range for all subjects.

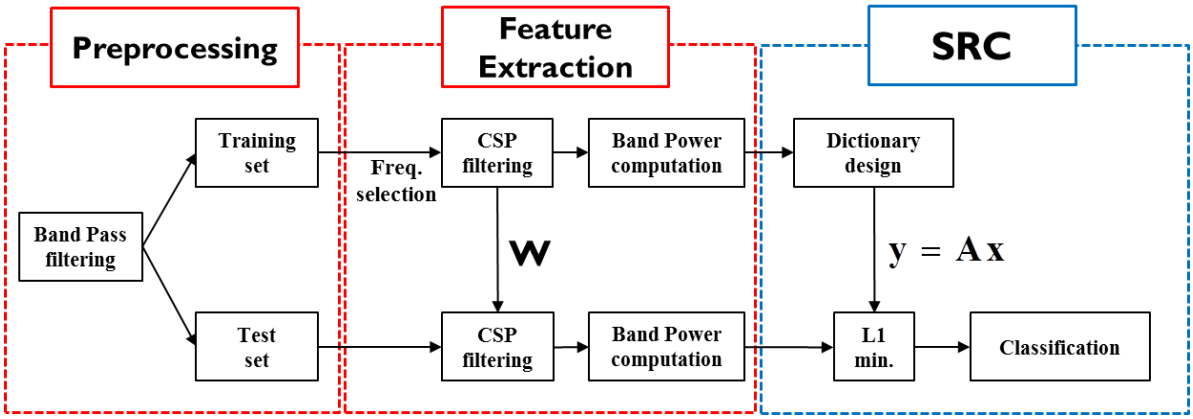


Figure 2.3 Proposed signal processing steps for SRC method

After band pass filtering, we reduce the dimension of EEG signal using the common spatial pattern (CSP) filtering, which is a widely used feature extraction method for motor imagery based BCI applications [5][6]. To find maximally distinguishable features, CSP filters maximize the variance of the spatially filtered signals for one class while minimizing it for the other class.

The CSP filtering algorithm finds the filters, $\mathbf{W} \in \mathbb{R}^{C \times C} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_C]$ which transforms the EEG data $\mathbf{X} \in \mathbb{R}^{C \times S}$ (C and S denote the number of EEG channels and time samples) into a spatially filtered space: $\mathbf{X}_{CSP} = \mathbf{W}^T \cdot \mathbf{X}$. Generally, $\mathbf{W}$ is computed by simultaneous diagonalization of the covariance matrices, Σ_1 and Σ_2, of the two classes of data. This is equivalent to solving the generalized eigenvalue problem, i.e., $\Sigma_1 \mathbf{w} = \lambda \Sigma_2 \mathbf{w}$, where λ is the eigenvalue. In practice, the first and last *n* columns of the $\mathbf{W}$ correspond to the *n* largest and *n* smallest eigenvalues that are used for CSP

filtering. However, the optimal number of CSP filters, $m = 2n$, which shows the maximum classification accuracy varies and has to be chosen empirically. Figure 2.4 shows eight CSP filters, $\mathbf{w}_1, \cdots, \mathbf{w}_4$ and $\mathbf{w}_{13}, \cdots, \mathbf{w}_{16}$, that correspond to the four largest and four smallest eigenvalues of subject *A* from the INFONET dataset.

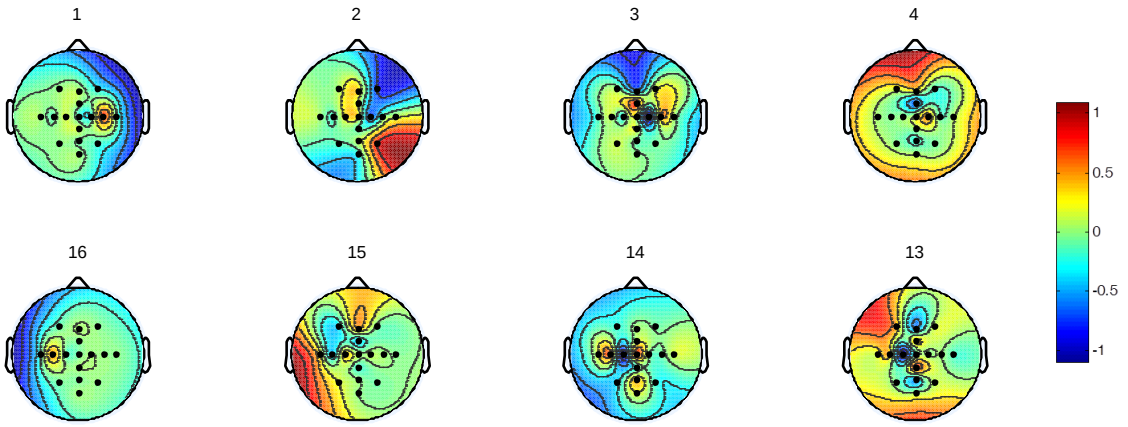


Figure 2.4 Color coded mapping of magnitudes of the coefficients of a filter, say $\mathbf{w}_i$ whose index *i* is indicated at the top of each figure. CSP filters are computed from the right-hand and the left-hand imaginary movement signals of subject *A* in INFONET dataset. The color map is projected onto the scalp.

The color in Figure 2.4 denotes the magnitude of the filter coefficients for the corresponding channels. For example, the red color indicates the highest significance. Let us take the upper left picture as an example, which is for the first filter $\mathbf{w}_1$. It is noticeable that there is a strong focus, a red dot, on the position of the C4 electrode in this case. Imaginary movement of a hand causes an ERD feature in the contralateral hemisphere. Namely, with a left-hand imaginary movement, a signal feature appears on the C4 electrode; while with a right-hand imaginary, it appears on the C3 electrode. Hand motor functions are controlled in a motor cortex region of the brain on which the C3 and C4

electrodes are placed. This contralateral manifestation of imaginary hand movements is a well-known neurophysiological phenomenon [2][7]. From this discussion, it is clear that the first CSP filter amplifies the signal feature from the left-hand imaginary movement at the C4 electrode; while the last CSP filter does the one from the right-hand imaginary movement.

After CSP filtering, we compute the band power (BP) of sensorimotor rhythm (8~15Hz). BP is the power of the signal within specific frequency bands. Because of the physiological background of the motor imagery signals, ERD based band power (BP) of the sensorimotor rhythm is a well-known feature form in many EEG based BCI studies [18][20].

2.4 SRC scheme for MI based BCI

2.4.1 Incoherent dictionary design using CSP filtering

An important step when applying the SRC method to the motor imagery BCI application is the design of an appropriate dictionary matrix, **A**. We use a CSP filtering technique to make maximally incoherent dictionary. Figure 2.3 shows how this is performed. Let N_t be the total number of training signals for each class, *i*. That is, *i* = *L* for the left-hand, *i* = *R* for the right-hand. We define a component dictionary matrix $\mathbf{A}_i = [\mathbf{a}_{i,1}, \mathbf{a}_{i,2}, ..., \mathbf{a}_{i,N_t}]$ for each class i where each column vector $\mathbf{a}_{i,j} \in \mathbb{R}^{m \times 1}$, $j = 1, 2, ..., N_t$, having dimension $m = 2n$ is obtained by concatenating the number of $2n$ SMR band power features, i.e., $2n$ sum of frequency power from 8 to 15 Hz. Here, $2n$ is the number of CSP filters. The same procedure is repeated for the left-hand and right-hand classes. By

concatenating the two matrices, we form the complete dictionary, $\mathbf{A} := [\mathbf{A}_L ; \mathbf{A}_R]$ as shown in Figure 2.5, where the dimension is $m \times 2N_t$.

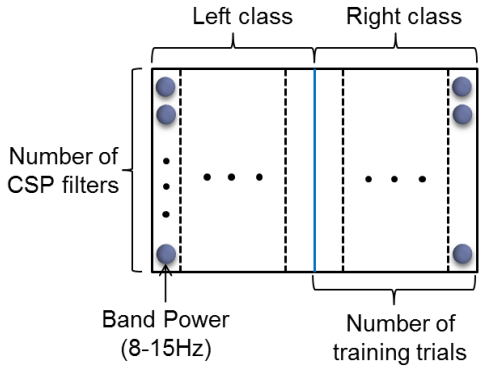


Figure 2.5 Dictionary design for proposed SRC method

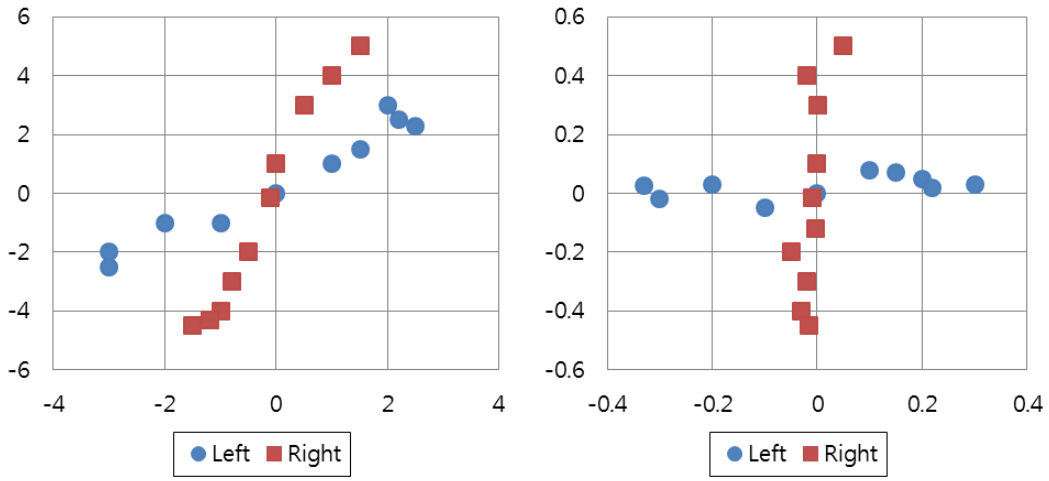
Now, we discuss why the CSP filtering method is a good technique to use for the design of the dictionary matrix. We also demonstrate how CSP filtering can be used to maximize the incoherence between the two classes in the dictionary.

The coherence measures the correlation between the two component dictionaries defined in the following way:

$$M(\mathbf{A}_L, \mathbf{A}_R) \triangleq \max \left\{ \left| \left\langle \mathbf{a}_{L,j}, \mathbf{a}_{R,k} \right\rangle \right| : j,k = 1,2,...,N_t \right\} \quad (1)$$

The vector $\mathbf{a}_{L,j}$ is the *j*-th column of $\mathbf{A}_L$; similarly, $\mathbf{a}_{R,k}$ is the *k*-th column of $\mathbf{A}_R$. The notation $\left\langle \mathbf{a}_{L,j}, \mathbf{a}_{R,k} \right\rangle$ denotes the inner product of two vectors. We call *M* the measure of mutual coherence of two component dictionaries; when *M* is small, we say that the dictionary is *incoherent* [21]. When a dictionary is *incoherent*, a test signal from one particular class can be predominantly represented by the columns of the same class. Therefore, sparsely represented test signal helps in

boosting the classification accuracy of the proposed method. The *uncertainty principle* (UP) [22] in the sparse representation theory dictates that a signal cannot be sparsely represented in both classes simultaneously. This phenomenon intensifies as the degree of incoherence of the dictionary increases.



(a) Before CSP filtering (b) After CSP filtering

Figure 2.6 Example of CSP filtering effect

Recall that we use the CSP filtering method. The CSP filter maximizes the variance of the spatially filtered signals for one class, while minimizing it for the other class. Figure 2.6 shows a two-dimensional example illustrating the effect of CSP filtering and its relation to incoherence. Two classes of samples are expressed by the blue circles and red squares. Figure 2.6 (a) shows the distribution of the samples before CSP filtering. Figure 2.6 (b) shows the distribution of the samples after CSP filtering. In (b), the horizontal axis is $\mathbf{w}_1$, which is an eigenvector corresponding to the largest eigenvalue. The vertical axis is $\mathbf{w}_{16}$, which is an eigenvector corresponding to the smallest eigenvalue. This $\mathbf{w}_1$ has the property that after the samples are projected onto $\mathbf{w}_1$, the variance of the projected samples for the left class (Blue) is maximized, while the variance of the projected samples

for the right class (Red) is simultaneously minimized. In addition, $\mathbf{w}_{16}$ does exactly the reverse of $\mathbf{w}_1$. Using the effect of the CSP filter, we simultaneously form maximally uncorrelated feature vectors between the two classes. Thus, if we calculate and compare the mutual coherence, *M*, between the two classes, before and after applying the CSP filtering, surely the mutual coherence after the filtering (Figure (b)) is smaller.

2.4.2 Sparse representation model and L1 minimization

Now, we introduce our sparse representation model using EEG training and test signals. First, we obtained the test signal, **y**, using the same procedure used to obtain the columns of dictionary **A** in section 2.4.1. Thus, a test signal was transformed into a vector, $\mathbf{y} \in \mathbb{R}^{m\times 1}$, via the processes of CSP filtering and band power computation. Therefore, the dimension of **y** was the same as the dimension of the columns in dictionary **A**. Our sparse representation model can be expressed by this formula:

$$\mathbf{y} = \sum_{i=L,R} x_{i,1}\mathbf{a}_{i,1} + x_{i,2}\mathbf{a}_{i,2} + \cdots + x_{i,n_t}\mathbf{a}_{i,N_t} \tag{2}$$

where $x_{i,j} \in \mathbb{R}, i = L, R,\ j = 1,2,...,N_t$ are scalar coefficients. We represent this using a matrix algebraic form:

$$\mathbf{y} = \mathbf{A}\mathbf{x} \in \mathbb{R}^{m\times 1} \tag{3}$$

In our sparse representation model, equation (3), the number of total training signals (the number of columns in **A**) was $2N_t$, which was much larger than the number of CSP filters (the number of rows, *m*, in **A**). Thus, the linear equation (3) is under-determined $(m < 2N_t)$. This problem can be solved using L0 norm minimization:

$$\min \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{4}$$

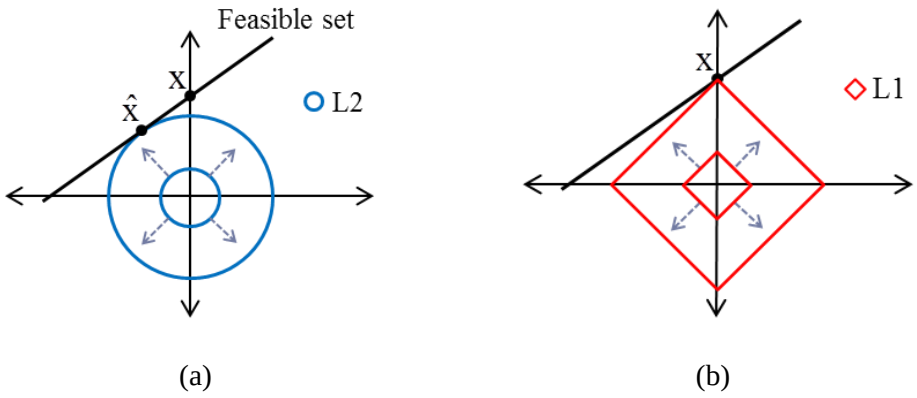
The L0 norm is equivalent to the number of non-zero components in vector **x,** by definition. This involves a combinatorial search. Therefore, solving this L0 norm minimization problem is an NP-hard problem. However, recent studies in compressive sensing (CS) theory have shown that signal reconstruction can be conducted using an L1 minimization technique [8][9]. This is a suboptimal and relaxed approach to the optimal but intractable L0 minimization. The L1 norm minimization, given below, as one of the big surprises in compressive sensing theory, finds the exact sparse solution in polynomial time [23]:

$$\min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{5}$$

unlike the conventional L2 norm minimization:

$$\min \|\mathbf{x}\|_2 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x} \tag{6}$$

Figure 2.7 shows a two-dimensional example of why the L1 norm minimization finds the sparse solution efficiently, unlike the L2 norm minimization. In Figure 2.7, the black line represents the set of all feasible solutions. From the definition of the norm, the L2 and L1 norms can be individually represented as vectors on the surface of the circle (blue) and rhombus (red) in (a) and (b), respectively. Using the L2 norm minimization in (14), when the L2 ball (circle) is grown in an equidistant manner, we can find the minimum L2 ball, which touches the feasible set first. As shown in (a), the L2 ball finds the non-sparse point, $\hat{\mathbf{x}}$, which lies in the two-dimensional non-zero space. Similarly, in (b), the L1 ball finds the sparse point, $\mathbf{x}$, which lies on the vertical axis.



(a) (b)

Figure 2.7 Visualization of L1 and L2 norm minimization.

There are many L1 minimization algorithms. In this study, we used a standard linear programming method called basis pursuit [24]. The "SolveBP" function implements the basis pursuit method available in SparseLab, which is a free MATLAB software package [25]. This function solves equation (5) by reducing it to a linear program using an optimization technique such as the primal-dual log-barrier algorithm.

2.4.3 SRC for MI signal classification

In the SRC algorithm, first, the columns of dictionary **A** are normalized to have a unit L2 norm. Then, the SRC scheme can be summarized in the following two steps. The first step is to sparsely represent a test signal **y** using the dictionary **A**. We called this step as a sparsification step. In this step, unknown coefficient vector **x** can be recovered by L1 norm minimization as shown in equation (5).

The second step is to determine class label of test signals. To make use of the sparse representation results in classification, we need to specify a classification rule. This step is called as the identification step. One simple method is simply to count the number of non-zero coefficients in vector **x**. Another

method is to compute the energy of the coefficients, i.e., $\text{Class}(\mathbf{y}) = \arg\max_{i=L,R} \|\mathbf{x}_i\|_2$. An additional effective classification rule is to use the minimum residuals as follows:

$$\text{class}(\mathbf{y}) = \min_i r_i(\mathbf{y}) \tag{7}$$

where $r_i(\mathbf{y}) := \|\mathbf{y} - \mathbf{A}_i\mathbf{x}_i\|_2$, $\mathbf{x}_i$ is the scalar coefficient vector corresponding to the class *i*. In this rule, we can check which class of training data can represent current test signal. We used this method as the classification rule in this study. Here, we summarize the SRC algorithm for two-class EEG signal classification.

1. Input: Collection of training features (dictionary) $\mathbf{A} := [\mathbf{A}_L; \mathbf{A}_R] \in \mathbb{R}^{m \times 2N_t}$, a test feature $\mathbf{y} \in \mathbb{R}^{m \times 1}$ where N_t is the total number of training trials for each class *L* and *R*.
2. Normalize the columns of $\mathbf{A}$ and $\mathbf{y}$.
3. Sparsification step: solve the following optimization problem

$$\min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}$$

4. Identification step: Compute $\min_i r_i(\mathbf{y})$ where $r_i(\mathbf{y}) := \|\mathbf{y} - \mathbf{A}_i\mathbf{x}_i\|_2$ for class $i = L, R$
5. Output: class label *i* of $\mathbf{y}$

Figure 2.8 shows the example of the sparse representation for two-class motor imagery based EEG signals and the classification rule. In this example, a certain test signal **y** of the right hand class can be predominantly represented with some training signals of the right hand class. This is represented by the nonzero scalar coefficients **x** in the position of corresponding class. However, EEG signals are very noisy and non-stationary in MI-based BCIs. Thus, non-zero coefficients may also appear in the

position of left-hand class. Using minimum residual rule in the identification step, we can classify the test signal.

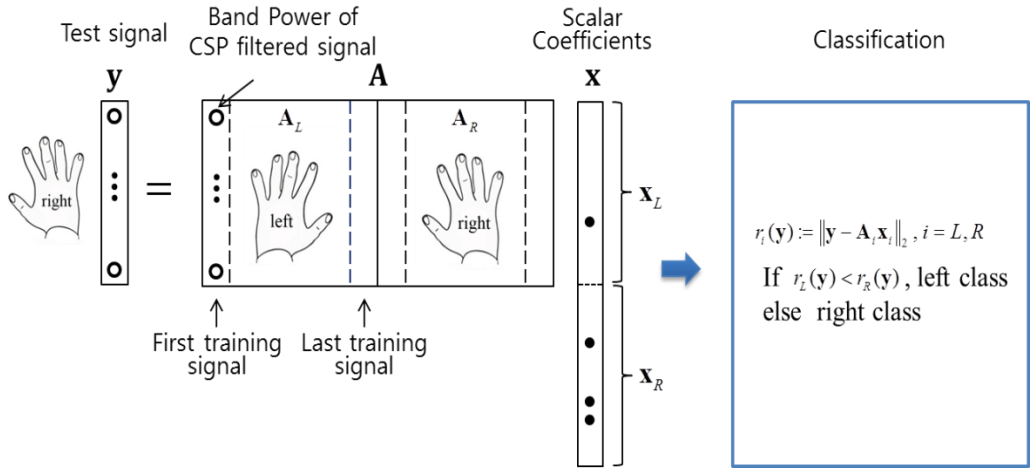


Figure 2.8 Example of SRC for motor imagery signal classification

2.5 Linear Discriminant Analysis

To provide a fair comparison between the SRC and LDA (also known as Fisher's LDA) classification methods, we aim to explain how the LDA classification method works when CSP filtering is incorporated. LDA is widely used as a linear classification method in the BCI field [6], e.g., see [26] for MI-based BCI applications.

The LDA approach introduced by Fisher aimed to find the optimal direction, $\mathbf{w}_L$, to project data upon and maximize Fisher's ratio: $J(\mathbf{w}_L) = \left(\mathbf{w}_L{}^T \mathbf{S}_\mathbf{B} \mathbf{w}_L\right) \Big/ \left(\mathbf{w}_L{}^T \mathbf{S}_\mathbf{w} \mathbf{w}_L\right)$ where $\mathbf{S}_\mathbf{B}$ and $\mathbf{S}_\mathbf{w}$ are called the *between-class* scatter matrix and *within-class* scatter matrix, respectively, which are obtained as follows: $\mathbf{S}_\mathbf{B} = (\mathbf{m}_2 - \mathbf{m}_1)(\mathbf{m}_2 - \mathbf{m}_1)^T$ and $\mathbf{S}_\mathbf{w} = \sum_i (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^T$ where $\mathbf{x}$ is the input feature vector, and $\mathbf{m}_i$ is the group mean of the feature vectors in class *i*. The logarithm of band power feature can be used for LDA classification. However, in this study, we used the band power of

the CSP filtered signals as a feature vector, **x**, which was exactly the same feature used in the proposed SRC scheme in Section 2.3 and Figure 2.5.

2.6 Results

In this section, we present the classification results with the proposed SRC method using the two datasets, INFONET and Berlin dataset, as described in Section 2.2. We also compare the results achieved with the SRC and the conventional linear discriminant analysis (LDA) method. To evaluate the average classification accuracy using limited size datasets, we used the statistical leave-one-out (LOO) cross-validation method [27] with the same total number of data trials for each subject. The LOO is useful for increasing the number of independent classification tests with a given set of limited data trials. Thus, one trial out of *k* total training trials is selected as the test trial, and the remaining *k*-1 trials are used as the training trials. This test is repeated for *k* times with different combination of training and test trials.

Table 2.1 Comparison of classification accuracy for INFONET dataset using SRC and LDA classification methods

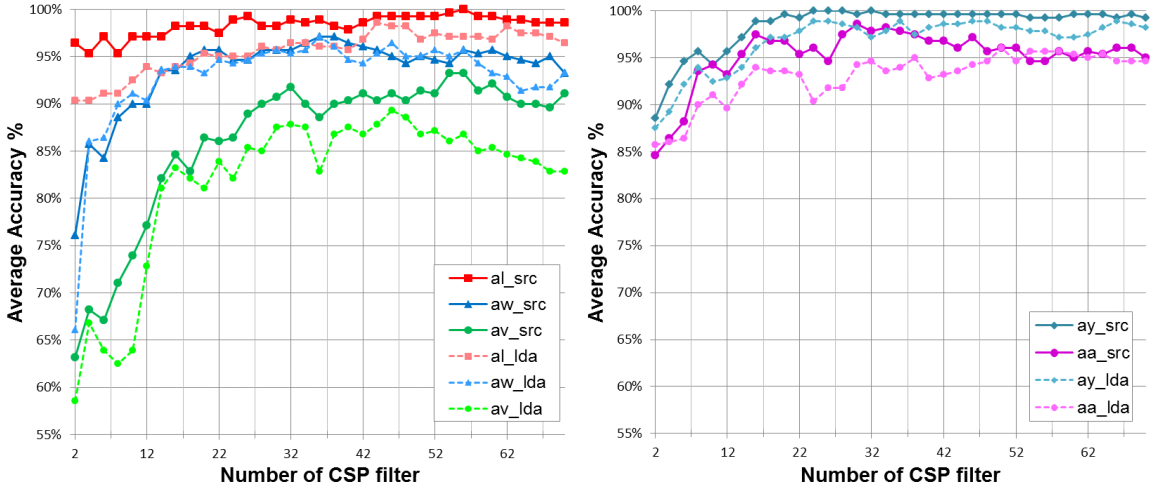
Subject	SRC Accuracy (%)	LDA Accuracy (%)
A	95.63	93.13
B	63.75	61.87
C	68.14	67.50
D	80	76.25
E	71.25	68.12
Mean (SD)	75.75 (12.60)	73.37 (12.18)

Table 2.1 shows the classification accuracy for the INFONET dataset. There are a total of 160 trial signals for each subject and therefore 160 LOO assessments are performed. We used the first and last CSP filters to produce the feature vectors and dictionary, i.e., $m = 2$ in (3), for subjects *A*, *B*, and *C*; whereas we used four filters, $m = 4$, for subjects *D* and *E*. The number of CSP filters to use was determined empirically. The results of Table 2.1 show that the proposed SRC scheme delivered enhanced classification accuracy compared with the conventional LDA method for all of the subjects.

Table 2.2 Comparison of classification accuracy for Berlin dataset using SRC and LDA classification methods

Subject	SRC Accuracy (%)	LDA Accuracy (%)
al	98.93	96.43
ay	100	97.14
aw	95.71	95.36
aa	97.86	94.64
av	91.79	87.86
Mean (SD)	96.85 (3.25)	94.29 (3.72)

To further evaluate the SRC method, we extended our validation to the Berlin dataset. This dataset was acquired from five subjects using 118 EEG channels. Table 2.2 shows the results of comparisons using this dataset. We used a total of 280 trial signals and the LOO method for all of the subjects. 32 CSP filters are used for feature extraction. Table 2.2 indicates that the proposed SRC scheme delivered higher average classification accuracy (96.85%) than the LDA method (94.29%). Moreover, for subject *ay*, the accuracy was 100%. Thus, the proposed SRC method had consistently higher classification accuracy than the LDA method in both datasets.



(a) (b)

Figure 2.9 Classification accuracy (%) per subject with different number of CSP filters for Berlin dataset. (a) Classification accuracies for subject *al*, *aw* and *av*. Solid line represents SRC results and dashed line represents LDA results. (b) Classification accuracies for subject *ay* and *aa*.

With the Berlin dataset, the number of available CSP filters was 118. We selected the number of CSP filters based on our experimental results in Figure 2.9. This figure shows the classification accuracy (%) of SRC and LDA as a function of the number of CSP filters for each subject. Figure 2.9 (a) shows the results of subject *al*, *aw* and *av*. Solid line represents the SRC accuracy and dashed line represents the LDA accuracy. Figure 2.9 (b) shows the results of subject *ay* and *aa*. We compute the average accuracy out of the 160 trials expressed in Figure 2.9 using the LOO cross validation method. As can be seen in these figures, there was no significant increase in accuracy when more than 32 CSP filters were used for both SRC and LDA methods. Thus, we used 32 CSP filters for feature extraction, which corresponded to the 16 largest and the 16 smallest eigenvalues. However, Figure 2.9 shows that for each selection on the number of CSP filters, SRC performs better than LDA does with few exceptions. Thus, it can be said that SRC has better classification accuracy than LDA regardless of the

number of CSP filters in Figure 2.9. To investigate the statistical significance of the observed accuracies in Figure 2.9, we performed a paired *t*-test for each subject. The obtained *p*-value of the *t*-test was less than 0.05 for all subjects, which indicates that the difference was significant.

An important issue for online BCI applications is the run time speed of the algorithm, as well as the classification accuracy. We compared the execution times of the algorithms. SRC and LDA took similar times to complete the classification job. The average difference between the execution times was negligible with the same computer and software (using MATLAB), i.e., LDA took 129.78 sec and SRC took 129.99 sec. The LDA shows 0.16% improved speed than the SRC and it is negligible for online BCI applications.

2.7 Summary

For the first part of this thesis, we applied the idea of sparse representation to the field of BCI systems and proposed a new classification method that delivered good performance for an MI-based BCI application. We used the well-known band power feature to utilize the event-related desynchronization (ERD) concept, which is the most widely used physiological feature in MI-based BCI application. This method required a well-designed dictionary matrix. We proposed a new procedure in which a CSP filtering technique is incorporated to produce the dictionary. We referred to this new classification system as the SRC method in this paper. To validate the SRC method, we applied it, not only to an INFONET dataset that we obtained in our laboratory but also to the publicly available Berlin dataset. In addition, we compared the proposed method with one of the well-known

classification methods, the LDA classification method. The results indicated that the proposed SRC scheme delivers classification accuracy higher than that of the LDA method. We noted that incoherently designed dictionary, together with the use of L1 minimization, makes SRC competitive as a classification tool.

3. Evaluation of SRC for non-stationary EEG signals

3.1 Motivation

It is well known that EEG signals are non-stationary. The non-stationarity can be observed during the change in alertness and wakefulness, eye blinking, and in the event-related potential (ERP) and evoked potential (EP) such as motor imagery signals. Because of the non-stationarity of the EEG, we can observe that the test feature positions vary from the original training feature positions in the feature space [6][28]. This is one of the major obstacles in EEG signal classification. Thus, a classifier that is optimized for a particular training data may not work for online BCI with a new test data.

Recently, extensive research efforts have been devoted to overcome the non-stationary issue in the motor imagery based EEG classification. Robust feature extraction methods were proposed for common spatial pattern (CSP) [29][30], which is the most widely used technique for feature extraction in the motor imagery BCI. In the classification stage, supervised and un-supervised adaptive classification schemes were studied for the conventional LDA and SVM methods [28][31][32].

In this Section 3, our aim is to evaluate the robustness of SRC for non-stationary EEG signal classification. First, we compare the classification performance, i.e., classification accuracy and computation time, of the SRC with SVM, which has been known as the state of the art classifier in many studies. Second, we evaluate the noise robustness of the SRC and SVM methods. For this purpose, we generate noisy test signals which have different feature distribution with original test

signals. The noisy test signals are generated with the addition of random Gaussian noise and scalp recorded background EEG signal into the original test signal. Then, we assess the noise robustness of both SRC and SVM methods. Third, in addition to the simple performance comparison, we examine working mechanism of SRC by analyzing advantages and disadvantages as the role of classifier compared with the conventional SVM. Moreover, we discuss why SRC outperforms SVM for the noisy test signal. Our work is intended to provide evaluation and analysis of SRC to researchers who want to apply the SRC framework to non-stationary EEG signal classification.

3.2 Experimental dataset

In Section 3, to evaluate and analyze the SRC method, we use two-class EEG based motor imagery experiment. This dataset is obtained from our collaboration laboratory, Biocomputing Lab., in GIST (Gwangju Institute of Science and Technology). Twenty healthy subjects (11 male and 9 female subjects whose average age is 24.05±3.76) participated in this experiment. Therefore, we collected 20 motor imagery datasets. Each dataset contains EEG signals generated from the left and right hand motor imagery experiment. Experiment included five runs. One run consisted of 20 trials for each class. Thus, the total number of trials was 100 for each instruction (class).

Figure 3.1 shows a single trial experimental paradigm of our motor imagery experiment. Cue line indicates the starting point of motor imagery. One trial consisted of 4–6 sec of resting time period and 3 sec of imagery time period. In the resting period, a blank screen appeared on the monitor. The resting time was randomly selected in the range of 4 to 6 sec. In the imagery period, one of the motor

imagery instructions was represented at the center of the screen, and then subjects imagined their left or right hand movements for tasks such as grasping and releasing hand. In each trial, instruction was randomly selected from the left and right hand class.

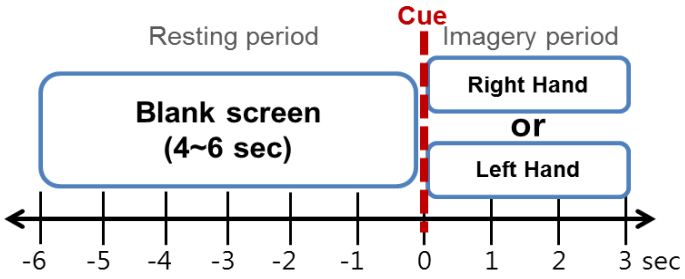


Figure 3.1 Single trial time procedure of motor imagery experiment

In addition, resting state EEG signals were recorded for each subject to estimate the subject-specific background noise. In this recording, subjects were instructed to open their eyes for 60 sec without any experimental task. These experimental datasets were recorded by an active electrode cap. We used Active Two EEG measurement system made by Biosemi, Inc. The sampling rate for these datasets was 512 samples per second, and the number of EEG channels was 64. The channel positions were selected from the international 10/20 standard.

3.3 Feature extraction and Classification

Preprocessing and feature extraction steps are common to both SRC and SVM classification algorithms. We apply same preprocessing and feature extraction methods to MI dataset as shown in Figure 2.3. Thus, we perform the band-pass filtering with a passband of 8–15 Hz. Then, we use the

CSP filtering and band power computation for feature extraction. For the SRC scheme, we use same procedure as explained in 2.4.3.

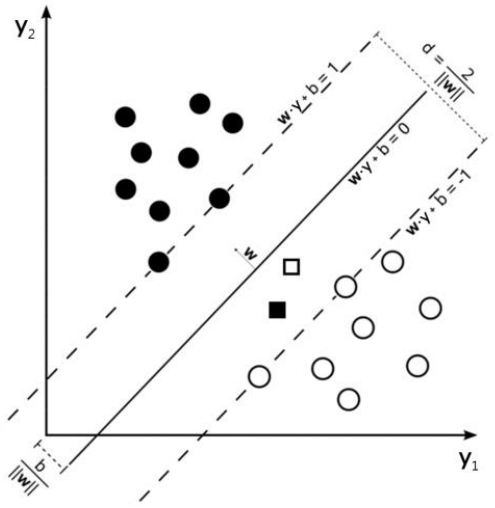


Figure 3.2 Main idea of SVM. The SVM algorithm tries to find the decision hyperplane, which has the maximum margin *d*.

To evaluate the noise robustness of SRC, we compare classification performance of the SRC with conventional SVM method. SVM is a well-known classification method in the area of pattern recognition and machine learning. In the BCI field, the SVM has shown a robust classification performance in many experimental studies [6][33]. SVM is recognized for its excellent generalization performance, i.e., small error rate for test data. This property is achieved through the idea of margin maximization. As shown in Figure 3.2, the margin d is twice the distance between the support vector (the black and white circles that are on the dashed line) and the decision hyperplane. The hyperplane can be described by a weight vector $\mathbf{w}$ and a bias b. The SVM finds the decision hyperplane by solving the following optimization problem:

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_n \xi_n, \\ &\text{subject to} \quad t_n(\mathbf{w}^T\Phi(\mathbf{y}_n) + b) \geq 1 - \xi_n \\ &\qquad\qquad \xi_n \geq 0, n = 1, ..., N \end{aligned} \tag{8}$$

where $\mathbf{y}_n$ is the training feature vector, $t_n \in \{+1, -1\}$ is the class information and n indicates the training trial number. To consider the training error, a slack variable ξ and a regularization parameter C are included. Using ξ, we can consider the training error which is positioned inside the support vectors. C is a user defined regularization parameter to control the importance between the maximum margin and the training error.

In the SVM optimization problem, mapping function $\Phi(\cdot)$ can be used to map an inseparable feature vector onto a higher-dimensional space using a kernel function $K(\mathbf{x}, \mathbf{y})$. In BCI research, the Radial Basis Function (RBF) kernel (4) is widely used and has shown robust classification performance:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{-\|x - y\|^2}{2\sigma^2}\right) \tag{9}$$

Therefore, in this study, we consider a linear SVM and an RBF kernel based SVM for comparison of the classification performance with the SRC method. For both SVM algorithms, we use the MATLAB Bioinformatics Toolbox (SVMtrain) [34].

In the SVM algorithm, selection of parameters is important to obtain the robust performance. We optimize the regularization parameter C in (3) for linear SVM and kernel parameter σ in (4) with combination of C for RBF SVM. We adopt a coarse grid search method using cross-validation to find

optimal parameters that provide the best classification accuracy [35]. In the exhaustive coarse grid search, we first find a better region on the loose grid, and then fine grid search on that region is conducted. For two parameters C and σ, we set the same grid sequence as follows:

C and $\sigma = [10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}]$. Then, for the best region, we optimize the parameters using a fine tuning.

3.4 Noise Robustness Analysis Method

In Section 3 of this thesis, we aim to evaluate the noise robustness of the SRC and SVM classification methods when test data is contaminated by an additive random Gaussian noise and scalp recorded background noise. The ultimate goal of this evaluation is to assess the classification performance of both methods for non-stationary EEG signals. As it is well known, EEG signals have inherent non-stationary characteristics. Therefore, BCI features vary from training sessions to test sessions during a BCI experiment [28]. There are many reasons to change EEG signals in the motor imagery task such as physical and mental drifts, misalignment of sensors, and task–irrelevant background activity [29][36]. During the imagery period in the motor imagery experiment, when we assume subjects exclusively perform motor imagery task, the task-irrelevant background activity can be the main reason for a change in EEG signals [28][29]. Therefore, in this context, we aim to model the modified noisy test signals by adding background activity estimated by the resting state recording into the original test signal.

For robustness analysis, we generate the modified test data by introducing two different noise sources such as white Gaussian and background noise into the original test data. Each noise source signal is separately applied to the EEG test data. Thus, we evaluate the classification performance of both classifiers for two types of noise corrupted test data.

Figure 3.3 shows the generation concept of the polluted noisy test data using one noise source. In the online BCI experiment, the power of EEG test data varies. Therefore, to evaluate the noise robustness of the classifiers systematically, we generate five different noisy test data with various SNR levels. Thus, we control the noise power of each noise source in five levels.

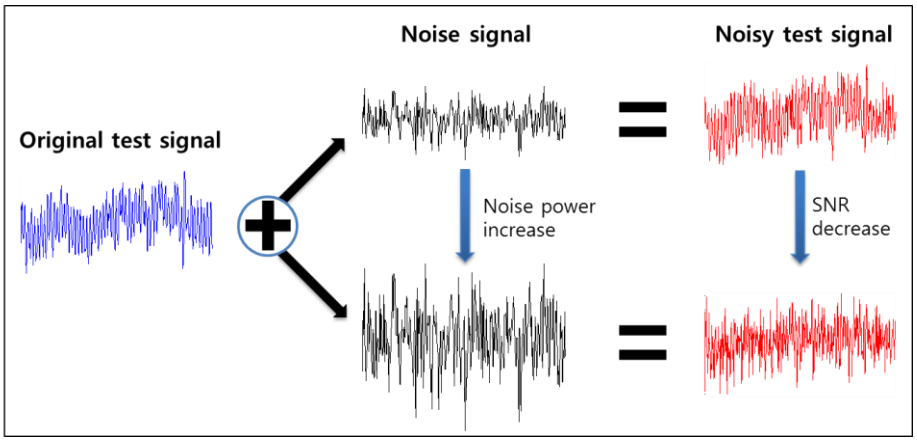


Figure 3.3 Noisy test signal generation using different power of noise signal

For the Gaussian noise, we control the noise power by varying the standard deviation of Gaussian distribution. For the background noise, we use a scale factor α to control the noise power as follows: $\text{polluted test signal} = \text{test signal} + \alpha(\text{resting noise})$. For each subject's dataset, the classification performance of the SRC and SVM methods is evaluated using both types of noisy test data.

Random Gaussian noise is artificially generated by 1-dimensional Gaussian distribution, i.e., $N(\mu, \sigma^2)$ where μ and σ^2 are the mean and variance. We use a MATLAB built-in function to generate the zero mean Gaussian distribution with different standard deviation σ. To make polluted EEG test data by Gaussian noise, we generate the same dimension of Gaussian noise to the segmented EEG signal, i.e., we generate 64 Gaussian noises which have 512 samples in each distribution. We also apply the band pass filter to the generated Gaussian noise with 8–15 Hz cutoff frequency, which is used in the preprocessing of EEG signal.

Subject-specific background noise is measured by the EEG recording of the resting state. In this recording, subject is instructed to just open their eyes without any task for one minute. We apply the band pass filter to the recorded resting state signal. To make polluted EEG test data by background noise, we collect one-second time samples (512 samples) from the resting state signal. In this study, both classifiers are evaluated using 100 test trials. Therefore, we generate 100 noise signals using the moving window from the total resting state signal. The size of the moving window is 256 samples (0.5 second).

3.5 Results

To evaluate and compare the classification accuracy of the SRC and SVM methods, we use the leave-one-out (LOO) cross-validation. Thus, one trial out of 100 training trials is selected as the test trial, and the remaining trials are used as the training trials. This test is repeated for 100 times with different combination of training and test trials. To obtain noisy test trials, we apply 100 different

noise signals for each noise source into the 100 test trials acquired from LOO cross-validation. Therefore, we have 100 noisy test trials for each Gaussian and background noise.

3.5.1 Comparison of Classification Results

First, we evaluate the classification accuracy of the SRC and SVM methods for the original experimental datasets that are not contaminated by noise sources. Figure 3.4 shows the comparison result of the classification accuracy for the SRC, linear SVM, and RBF SVM. For each subject, we computed the classification accuracy (in %) using the LOO cross-validation. We used 18 CSP filters for both classification methods, which are determined heuristically (see Figure 3.5).

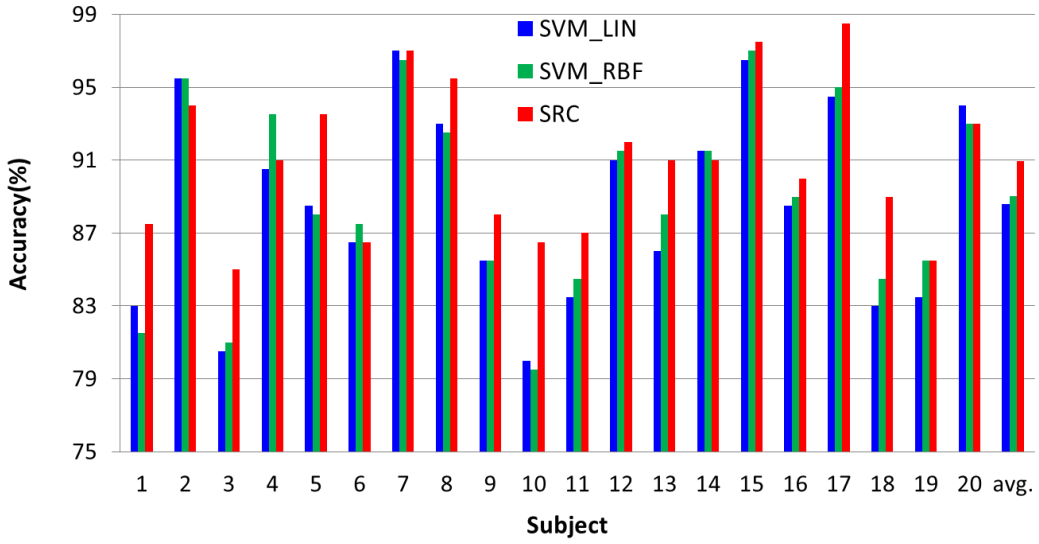


Figure 3.4 Comparison of classification accuracy for the linear SVM, RBF kernel SVM, and SRC method using 20 non-noisy experimental datasets

In Figure 3.4, we observe that SRC achieves competitive classification accuracy over both linear and RBF kernel-based SVM. The classification accuracy of SRC was found to be better than linear SVM for 15 subjects and RBF SVM for 14 subjects over 20 subjects. In addition, the mean difference

of the classification accuracy between the SRC and both SVM methods was statistically significant using the paired t-test ($p < 0.01$).

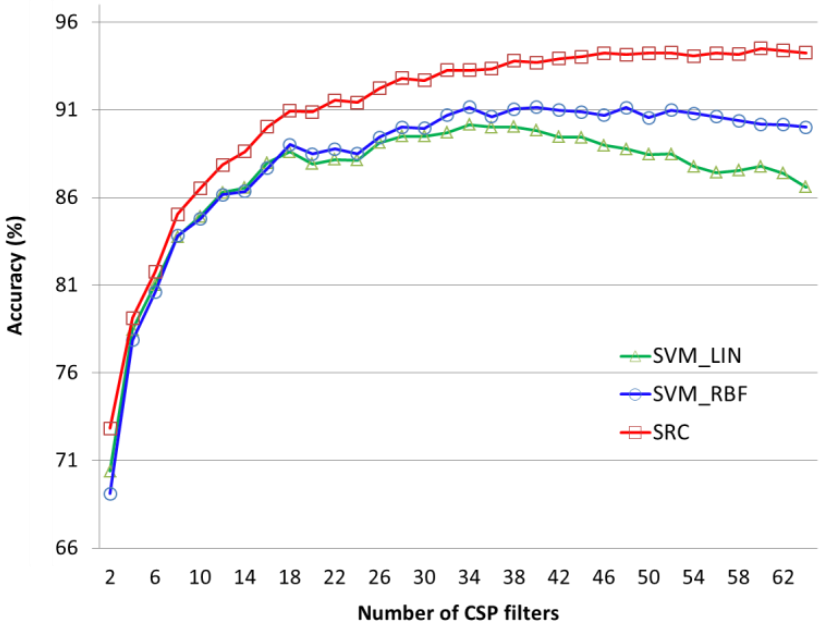


Figure 3.5 Average classification accuracy over 20 non-noisy datasets when the number of CSP filters (feature dimension) is varied from 2 to 64.

Furthermore, we investigated the impact of varying the feature dimension on the non-noisy classification performance in Figure 3.5. In this study, we used the CSP filtering as a feature selection method. The number of CSP filters (feature dimension) was varied from 2 to 64. Usually, the optimal number of CSP filters, which showed the maximum classification accuracy was chosen empirically. However, the optimal number of CSP filters was different depending on the classification method and dataset. Therefore, we evaluated the classification performance of each classification method when the feature dimension was varied. Figure 3.5 shows the average classification accuracy over all subjects when the number of feature dimensions m was varied from 2 to 64. We found that the classification accuracy of SRC method consistently outperformed the linear and RBF kernel based

SVM methods regardless of their feature dimension. There was not much difference in the classification accuracy between the SVM methods. However, the RBF SVM showed better classification accuracy when the number of CSP filters was over 18. We used the fixed 18 CSP filters for all classification methods that are shown in Figure 3.4. However, the result in Figure 3.5 shows that this number was not optimal for certain classification methods. When we used more CSP filters, the difference in the classification accuracy between the SRC and SVM methods was increased.

3.5.2 Classification Results for Noise Robustness

In this section, we evaluate noise robustness of the RBF kernel based SVM and SRC methods. For the noise robustness analysis, we used polluted test signals that were generated by adding two noise sources, i.e., white Gaussian noise and background noise, into the original test signal.

Figure 3.6 shows the noise robustness results of the SRC and RBF kernel based SVM methods for the Gaussian noise. The average classification accuracy over all subjects was assessed when the noise power was varied. For the Gaussian noise, we controlled the noise power by changing the standard deviation, and the SNR was computed for different noise powers. In this study, SNR computation was defined as follows: $\mathrm{SNR(dB)} = 10\log_{10}\left(P_S / P_N\right)$ where P_S and P_N indicate the signal and noise power, respectively. For the SNR computation, we investigated the average SNR over all the channels and subjects. As shown in Figure 3.6, we found that the classification accuracy of SRC was higher than that of the RBF SVM for all SNR cases. The difference in the classification accuracy between the SRC and RBF SVM was increased with the SNR increase.

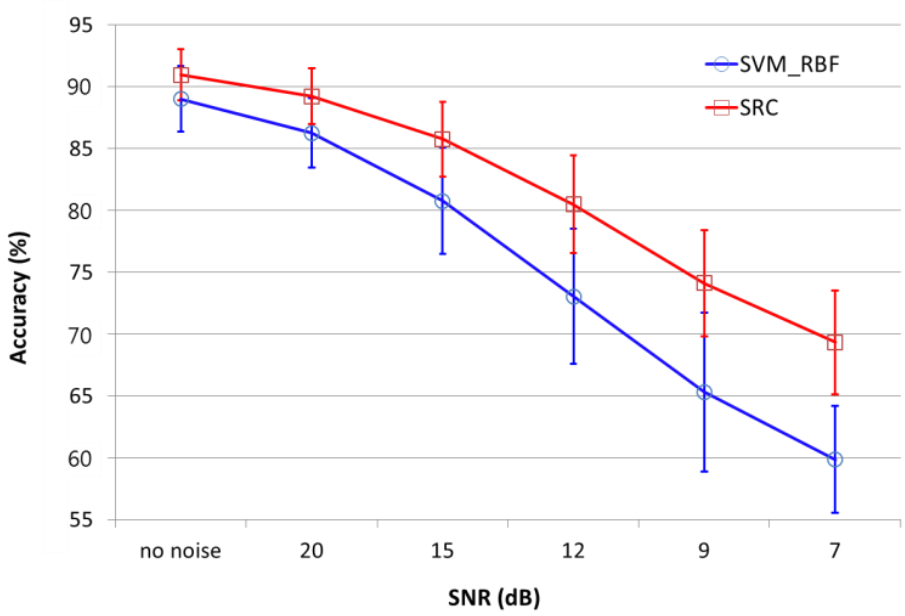


Figure 3.6 Comparison of the average classification accuracy over 20 subjects. Average classification accuracy for Gaussian noise is represented as a function of SNR. Vertical line indicates the standard deviation of the accuracy for each SNR.

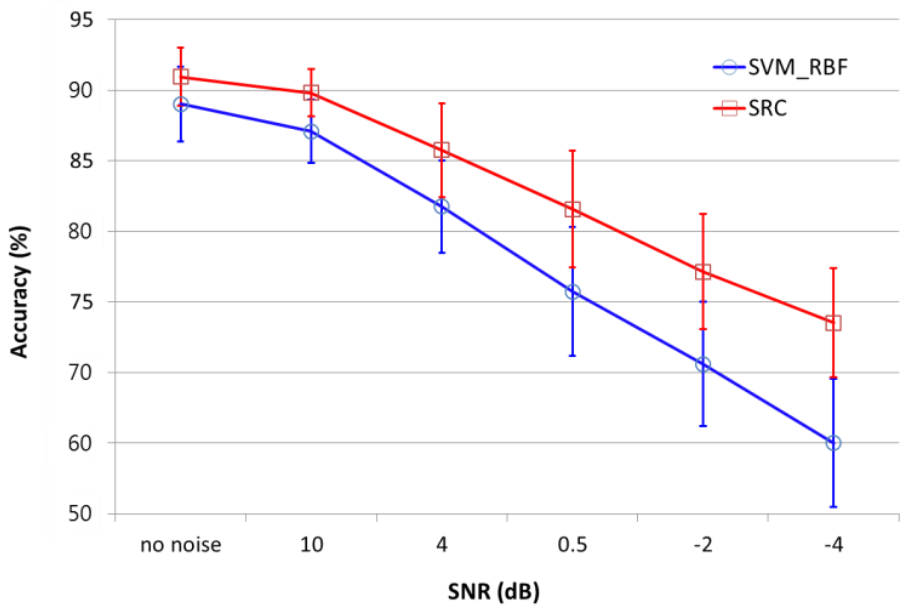


Figure 3.7 Comparison of the average classification accuracy over 20 subjects. Average classification accuracy for background noise is represented as a function of SNR.

Similarly, Figure 3.7 shows the noise robustness results of the SRC and RBF kernel based SVM methods for the background noise, which was measured by the recorded resting state. For the background noise, the noise power was controlled by scale factor α (see Section 3.4). It was found that the classification accuracy of SRC was higher than that of the RBF SVM for all SNR cases. In addition, when the noise power increased, the accuracy difference between the SRC and SVM increased. For example, in the noiseless case, the average accuracy difference between the SVM and SRC was 1.9%. However, in the case of 0.5 and -4dB SNR, the difference was 5.8% and 8.5%. This means that the SRC method was more robust than the SVM for the polluted test signal in the background noise case.

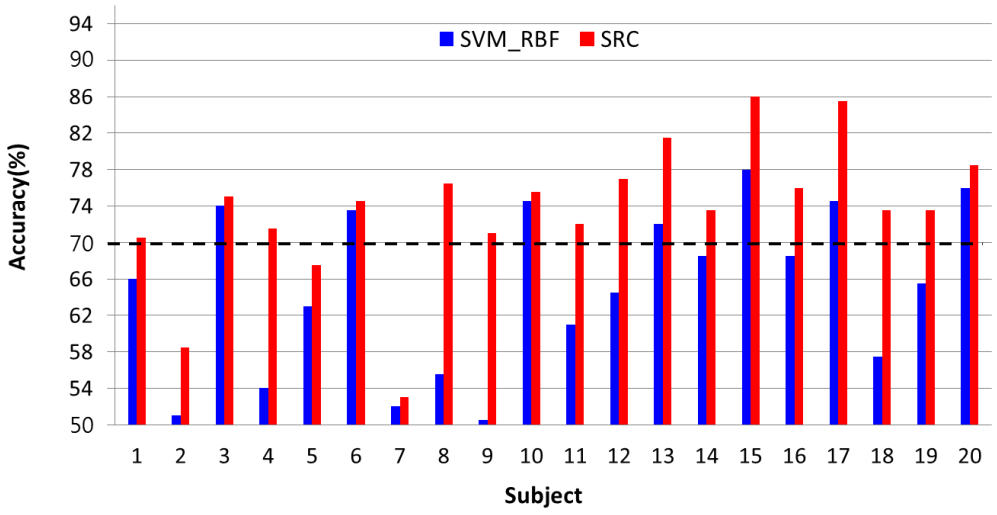


Figure 3.8 Classification accuracy of RBF based SVM and SRC method for polluted test data by background noise (-4dB).

In two-class classification problems, the theoretical chance level is 50%. However, in many EEG based BCI studies [37] [38], at least 70% classification accuracy is considered as a threshold for an acceptable communication and device control. In Figure 3.8, we examine the classification

performance for the polluted test data. Figure 3.8 shows the classification accuracy of all subjects for the -4dB SNR for background noise cases shown in Figure 3.7. The threshold of 70% classification accuracy is represented by black dotted line. For this threshold, the SVM has seven datasets that are over the threshold and the SRC has seventeen datasets. This means that for the noisy test data, 10 more subjects can use a reliable BCI system with the SRC compared to the SVM method.

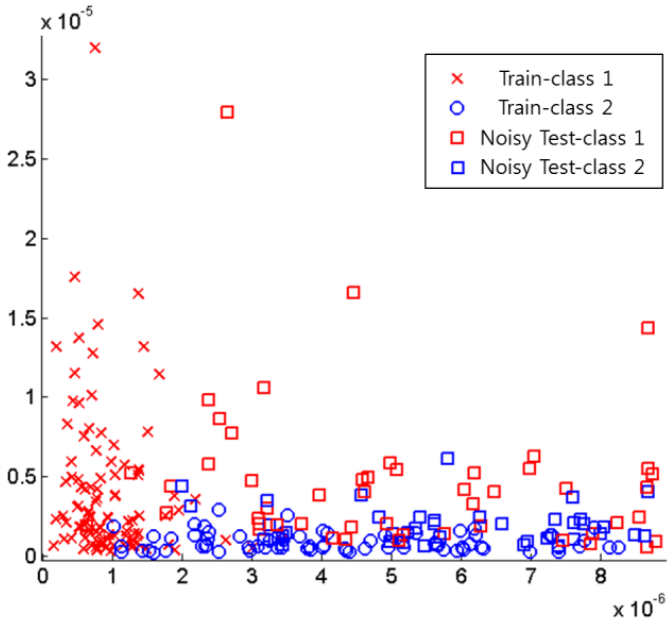


Figure 3.9 Scatter plot of training data and noisy test data in two-dimensional feature space (2 CSP filters) for one subject dataset. Noisy test data are generated using background noise with 4 dB SNR.

Figure 3.9 shows an example of training and polluted test features for one subject dataset. In this example, the background noise with 4 dB SNR (shown in Figure 3.7) was used for the polluted test data. The positions of noisy test features (red and blue squares in Figure 3.9) in two-dimensional feature space were relocated from the positions of the original training features (red x-marks and blue circles) to places with a particular direction. This represents a typical situation that occurs in real-time

BCI scenario where the online test data has different background noise compared to the training data. In this study, the positions of the noisy test features were varied according to the SNR of the test data.

3.6 Discussions

3.6.1 Comparison of Classification Mechanism

In this section, we examine the algorithmic difference between the SRC and SVM methods as the role of signal classification. Figure 3.10 shows the classification algorithms for both methods. Feature vectors for the training data were used as an input for both classification algorithms.

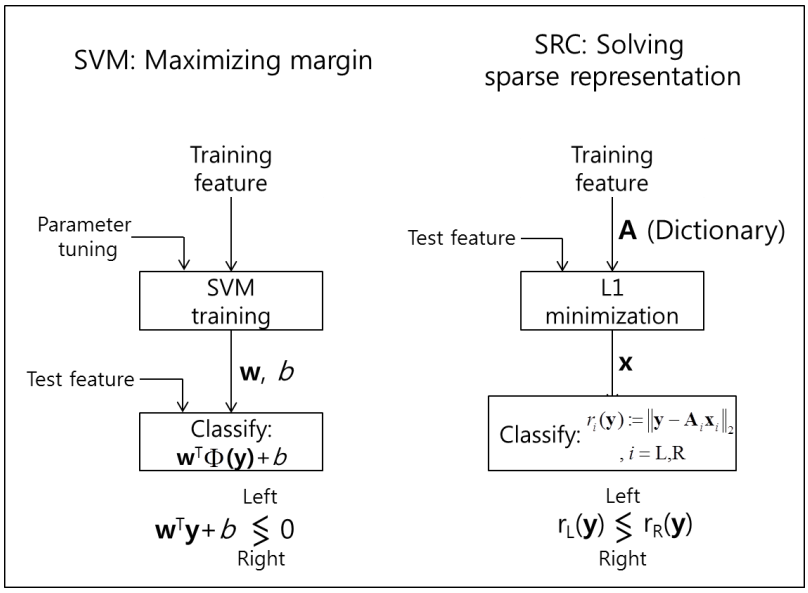


Figure 3.10 Comparison of the SVM and SRC classification algorithm

In the SVM algorithm, the input feature data and model parameters were used and the training was performed to find the parameters **w** and *b* for decision boundary as shown in Equation (8). Based on the boundary, the test feature was classified. Thus, the **y** class information was determined by the decision boundary.

On the other hand, in the SRC algorithm, the dictionary was simply formed by collecting the input training feature vectors as the columns of the dictionary. Then, using the dictionary, sparse representation was performed for each test data. Thus, scalar coefficient vector **x** was obtained by solving L1 minimization as shown in Equation (5). Using recovered coefficient **x**, class information was determined by computing the residual $r(\mathbf{y})$ in Equation (7).

We aim to highlight the important difference of the classification mechanism of the SRC and SVM methods as follows:

SVM (or LDA)	A fixed decision rule (decision boundary) was obtained for the entire set of training signals. Then, for each test signal, this fixed decision rule was used for signal classification.
SRC	The sparse representation was adaptively performed for each test signal by utilizing all training signals in the dictionary.

3.6.2 Robustness Analysis of SRC

The experimental results presented in Section 3.5 shows that SRC had a better classification accuracy than the conventional SVM for motor imagery based EEG signal. In addition, SRC was more robust for polluted test data than SVM. In this section, we discuss the relationship between the classification performance and the difference in the classification mechanism for SRC and SVM methods.

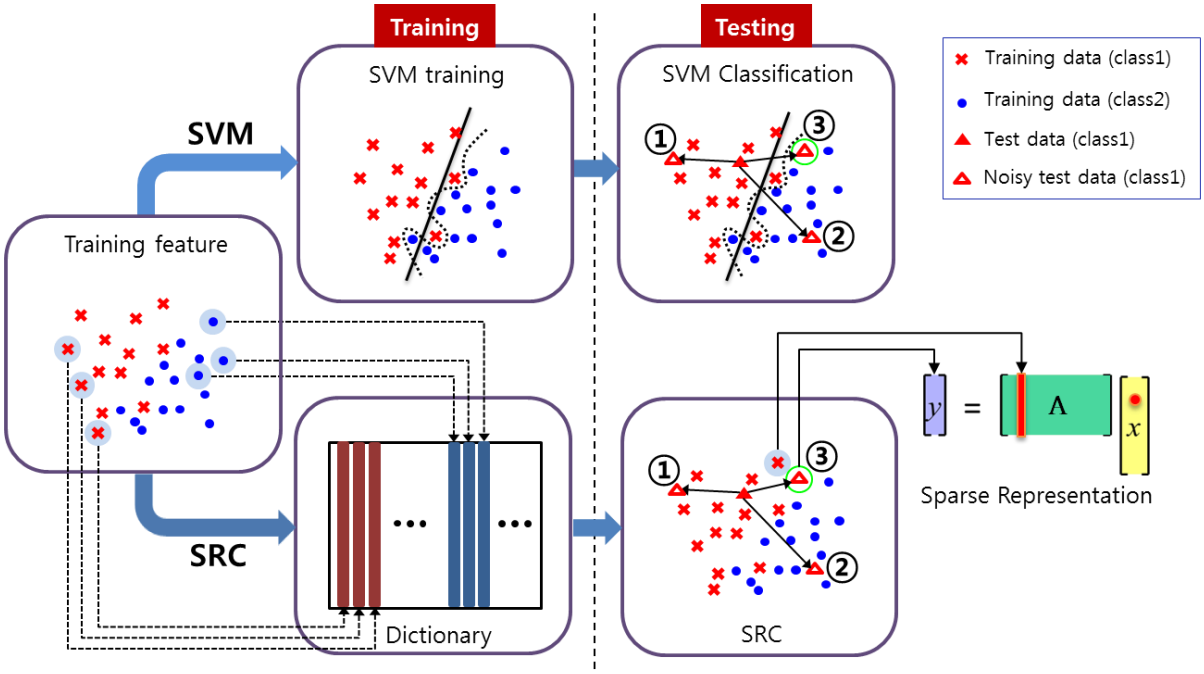


Figure 3.11 Comparison of the classification procedure and characteristic of the SVM and SRC for the noisy test data. In the SVM part, black solid line and black dotted line indicate the decision boundaries for linear and RBF based SVM.

Figure 3.11 shows the concept of the classification strategy for the SVM and SRC using a toy example of polluted test data in two- dimensional feature space. In the SVM classification, decision hyperplane and non-linear decision boundary were presented for linear and RBF based SVM. For many conventional classifiers including SVM, the classifier was trained using training data; thus, the best decision rule was determined. Then, this classification rule was applied to each test data. However, as we have shown in Figure 3.11, when the test data was polluted and shifted in feature space, the decision rule could not guarantee a satisfactory classification performance. On the other hand, in the SRC method, no classification rule was designed in the training part of SRC. Instead, a dictionary was formed by collecting feature vectors of the training data. Then, the sparse representation was performed for each test data using the dictionary. In addition, for the noisy test

data, an independent classification task was performed in each classification by using all the training data instead of a fixed decision rule.

For a detailed analysis, we considered three possible cases of polluted test data that are presented by numbers ①, ②, and ③ in Figure 3.11:

In the first case, test data was shifted away from the decision boundary and positioned at the same class feature space. In this case, both SVM and SRC correctly classified the noisy test data.

In the second case, the test data was positioned at a different class feature space of training data. Then, based on the decision boundary, the SVM classified the test data incorrectly. In the SRC method, the test data was more likely to be represented with different class training data. Thus, both classifiers were not working correctly.

Note that in the third case, similar to the second case, the test data was placed at a different class feature space. At the same time, the test data could be possibly positioned near the decision boundary. Usually, classification performance of classifiers can be determined by the test data of this case. Based on the decision rule obtained from the training data, the SVM resulted in wrong classification. When we used non-linear decision boundary, e.g., RBF SVM, as shown in black dotted line, this line was optimal for the training data. Thus, the classification error could be less than the linear decision hyperplane. However, for the polluted test data, the non-linear decision boundary was fixed.

On the other hand, in the third case, SRC still had a chance for correct sparse representation with the same class training data as shown in Figure 3.11. This was possible because the SRC method did

not depend on a fixed decision rule that was obtained from the training data. Instead, for each classification of test data, the SRC method directly used all training data and performed sparse representation.

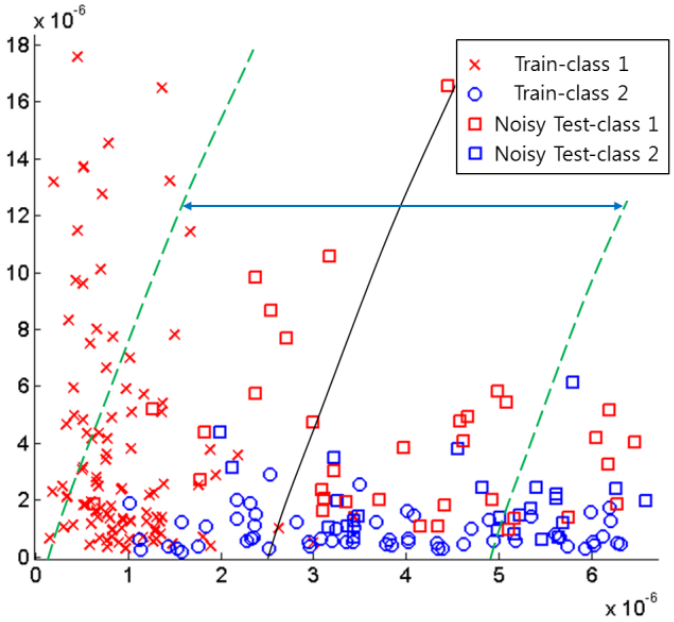


Figure 3.12 Scatter plot of training data and noisy test data in two-dimensional feature space for one subject data. Noisy test data are generated using background noise with 4 dB SNR.

To evaluate the validity of our analysis, we examined the same data shown in Figure 3.9 in details. Figure 3.12 shows an enlarged version of the scatter plot using the training and noisy test data. The black line indicates the obtained decision boundary from the RBF kernel based SVM. The region between the two green dotted lines is chosen as the near area of the decision boundary. In this area, many miss-classification cases may occur for both classifiers. In addition, most of the polluted test data, which correspond to case ③ in Figure 3.11 are located in this region.

For all noisy test data (i.e., 100 trials), the RBF SVM and SRC showed the classification accuracy of 56% and 62%, respectively. Because we used only two CSP filters for visualization, the classification accuracy was very low compared with the results given in Figure 3.7. For the noisy test data, which are located between the green dotted lines, the RBF SVM showed 57% classification accuracy. However, the SRC showed an improved classification accuracy of 83%. In addition, when we only considered the noisy test data for case ③ examples, the RBF SVM had 18 miss-classification data. However, the SRC correctly classified 12 test data among 18 test data. Therefore, we confirmed that the noisy test data of case ③ were miss-classified from the fixed rule based SVM. On the other hand, for the same data, the SRC correctly classified many times with the effort of independent classification task for each test data using all training data.

Figure 3.13 shows one instance of the noisy test data that was not correctly classified by the SVM; however, was correctly classified by the SRC method. The test signal of class 1 is represented by a red square, which is located in the region between the green dotted lines shown in Figure 3.12. The figure inside the green box shows the recovered coefficient **x** from the SRC method. Using the trial numbers (x-axis of the figure inside the green box) with large coefficient values, we represented the corresponding trials by the black x-marks and circles in Figure 3.13. Four largest coefficient values were selected for class 1. Two largest coefficient values were selected for class 2. As it can be seen, the noisy test trial of class 1 (red square) is located near the training trials of class 2. However, in the SRC method, using the coefficient **x,** the test trial could be correctly classified from the minimum residual rule in Equation (7). In addition, in each test trial, a different coefficient **x** which represented

the test data most compactly, was recovered by L1 minimization. Therefore, for the case of time varying EEG signal classification, the SRC approach was much more appropriate to employ than the SVM method, which was based on the fixed decision rule.

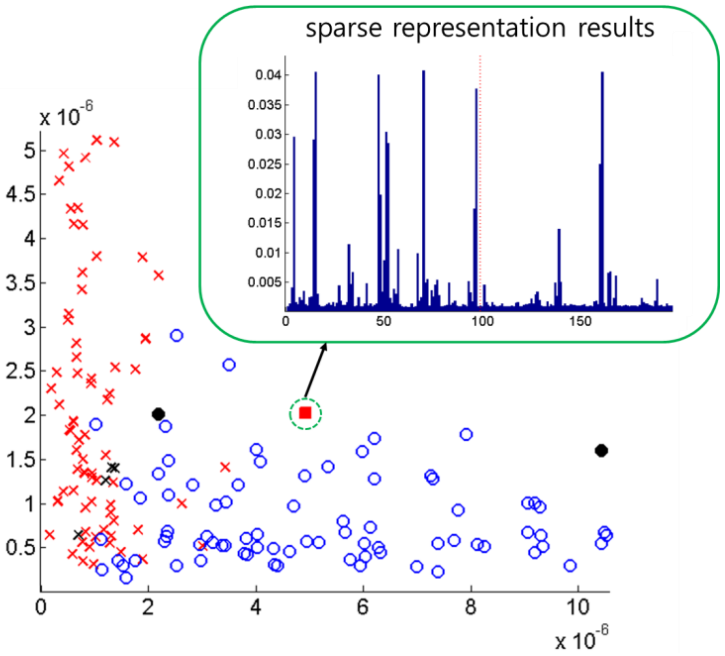


Figure 3.13 Scatter plot of training data and noisy test data. The figure inside the green box indicates the sparse representation result of the noisy test data.

3.6.3 Computation Time Analysis

In this section, we evaluate the computation time (running time) of the classification algorithms for the experimental datasets. As it can be seen in Figure 3.11, the most time consuming process of the SVM occurred while training the SVM. On the other hand, the most computation cost in the SRC algorithm occurred in L1 minimization step for sparse representation. Therefore, our evaluation for running time focused on the SVM training and L1 minimization step for the SRC algorithm. We used the *tic* and *toc* MATLAB commands to measure the start and end time of the SVM and SRC

algorithms, respectively. We simulated all algorithms in the same environment using MATLAB 7.14 (R2012a) with 3.30 GHz processor and 8 GB memory.

In the case of online BCI classification, typically the SVM decision boundary was designed once using the training data. Then, all the test data was classified based on the decision boundary. On the other hand, independent classification task was performed for each test data in the SRC. Therefore, the computation time of the SRC is increased by the number of test trials. Thus, a robust classification performance of SRC included the cost of the computation time at each test trial. For a single test trial, the average computation time for the SVM and SRC was 12.1 msec and 16.7 msec respectively. This computation time was averaged for 100 test trials of all subjects. The SVM shows 38% improved speed than the SRC. However, the difference is 4.6 msec and it is negligible for online BCI applications.

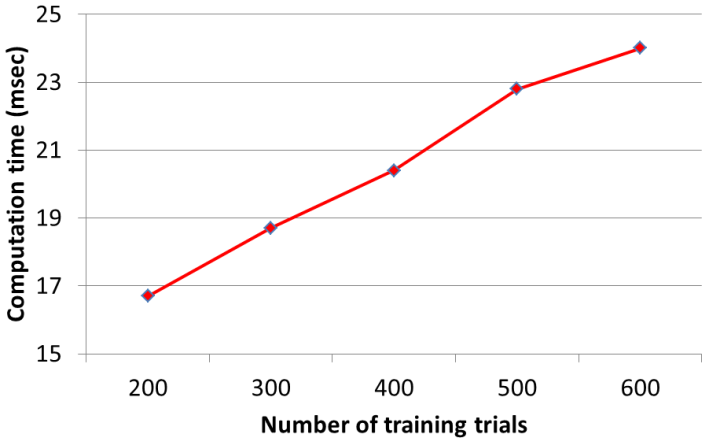


Figure 3.14 Computation time of the SRC as a function for the number of training trials.

In this study, the size of the dictionary, i.e., the number of training trials, was 200. In Figure 3.14, we display the average computation time as a function of the number of training trials. When the size

of dictionary was increased, the difference of the computation time was just a few milliseconds. Therefore, this was not an important factor for a single trial classification in online BCI systems. In addition, recently developed fast L1 minimization algorithms can be used for the SRC method. In [39], authors showed that some fast L1 minimization algorithms provided faster computation time than the conventional SRC method for large datasets of real face images. In addition, note that even though the computation time of the SVM was smaller than the SRC, the SVM required more effort to select a proper kernel and tune the model parameters for accurate classification results.

3.7 Summary

In Section 3, we evaluated and analyzed the robustness of the SRC method against the non-stationarity of EEG signal classification. For this purpose, we generated noise corrupted EEG test signals using two noise sources such as random Gaussian noise and scalp recorded background noise. Then, we assessed the classification performance of the SRC when the noise power was varied. Using the experimental motor imagery based EEG and generated noisy test data, we compared the classification results of the SRC with that of the SVM method, which has been considered as a robust classifier in many BCI studies. From the results, it was evident that the SRC showed superior noise robustness than the SVM for both Gaussian and background noise. We analyzed that the robust classification accuracy of the SRC was due to a different classification approach compared with the conventional decision rule based SVM. Thus, the SRC showed an inherent adaptive classification mechanism for each test trial via optimal sparse representation of the training trials. In addition, we

showed that the computation time of the SRC for a robust classification was on the order of milliseconds, which was acceptable for real time BCI systems.

4. Simple adaptive SRC schemes

4.1 Motivation

In the beginning of BCI research, BCI systems have been developed mostly to provide alternative communication means to people who have severe motor disabilities [2][7]. Recently, much research effort focused on development of portable BCI systems for normal person by using headset shaped scalp electrodes [40][41] and also dry electrodes which not need conductive gel for preparation of EEG recording [42][43]. In addition, with the progress of portable BCI systems and EEG sensor technologies, many BCI applications are developed for general public [43][44]. However, for the BCI systems going beyond laboratory researches, the most important issue is stable classification performance.

Normally, EEG based BCI experiment can be categorized as a training (calibration) stage and a real time testing (feedback) stage. In the training stage, translation algorithm such as classification is designed using collected training signals. Then, an application device such as neural prosthesis is controlled by using the classification algorithm in real time testing stage. However, EEG signals have inherent non-stationary characteristics and there exist significant day-to-day and even session-to-session variability [5][29]. Thus, features of experimental EEG signals are changed from the offline training sessions to online testing sessions [28]. Due to this, classification performance is unavoidably deteriorated in BCI experiment with time. This is one of major obstacles of real-time online BCI applications.

To overcome the performance decrease caused by the non-stationarity of EEG signals, many adaptive signal processing methods are proposed. In [29][30], adaptive feature extraction methods are proposed for the motor imagery based BCI systems. For the adaptive classification scheme, in [45], mean and covariance matrix of a statistical classifier are iteratively updated using each class data. The study [28] proposes a bias adaptation scheme of linear discriminant analysis (LDA) classification using class labels of several test trials. They have shown that simple bias adaptation is effective for online test data. In [46], they propose an expectation-maximization (EM) algorithm based unsupervised adaptive classification method. Using EM algorithm, common spatial pattern (CSP) features are re-extracted and parameters of Bayes classifier are updated in each iteration step. Similarly, [32] suggest unsupervised bias adaptation of LDA without using class label information. Previous studies for adaptive classification method need classifier re-adjustment (training) such as parameters and bias adaptation for new test trials. However, for this re-training, additional computation is needed in each update (adjustment) step.

Compared to other fixed decision rule based classification method such as linear discriminant analysis (LDA) and support vector machine (SVM), in the SRC, the sparse representation is adaptively performed for each test data by utilizing all training data in the dictionary (see Section 3). Along with this inherent adaptive characteristic of the SRC, in this study, we propose simple adaptive SRC schemes for real-time BCI applications. We suggest a dictionary update rule and an incoherence based dictionary modification (IDM) method. For the dictionary update rule, supervised and unsupervised adaptive schemes and also accumulated and fixed update rules are considered. Proposed

dictionary update methods are very simple and additional computation for adaptation is not needed. In the part of IDM method, our aim is to create a maximally incoherent dictionary via an incoherence measure of training data. This method is applied to the training data before performing the sparse representation. Using online motor imagery based BCI experimental datasets, we evaluate classification performance of the proposed adaptive method by comparing with the conventional SRC and other adaptive classification methods.

4.2 Experimental Dataset

In Section 4, for evaluation of adaptive classification schemes, we use an online motor imagery based BCI experimental dataset. This dataset is obtained from the collaboration laboratory, Biocomputing Lab., in GIST (Gwangju Institute of Science and Technology). The experiment was approved by the Institutional Review Board of GIST. Ten subjects who signed a written informed consent letter participated in our online experiment. The experiment was performed on multiple days (two or three days). In each day, just one session experiment was executed. The number of sessions for each subject was determined by classification results and condition of each subject. Right hand (R), left hand (L) and foot (F) motor imagery were performed for each subject. For this experiment, we used Active Two EEG measurement system made by Biosemi, Inc. The sampling rate of these datasets was 512 samples per sec and the number of EEG channels was 64. The channel positions were selected from international 10/20 standard.

The detailed experimental paradigm was illustrated in Figure 4.1. The same paradigm was used for both training (calibration) and online testing (feedback) phases. In the training phase, one session consisted of three runs and one run consisted of 20 trials for each class. Thus, we collected a total of 60 training trials for each class. All participants were naïve subjects for this motor imagery experiment. Therefore, it was difficult to achieve satisfactory classification performance without sufficient training time. In addition, each subject had a different discrimination potential for a different pair of motor imagery signals. In this study, to find the most discriminative motor imagery pair for each subject, we performed the initial classification for all pairs of (R), (L), and (F) by using the dataset of the first run in the training phase. The best pair of motor imagery was selected using the band pass filtered CSP feature (5-30 cut off frequencies and 10 CSP filters were used) with the LDA classifier and used for a further experiment in the training and testing session.

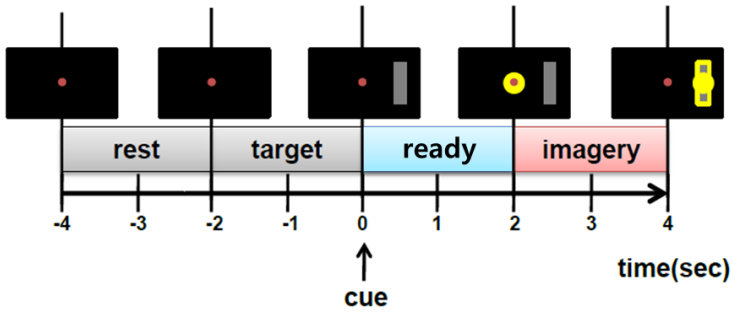


Figure 4.1 One trial time procedure of online motor imagery experiment

As shown in Figure 4.1, in each trial, the target bar was represented on zero sec at left, right or down side of monitor screen corresponding to the left, right or foot motor imagery. On two sec after cue onset, subject was instructed to perform the motor imagery task. Then, subject imagined their left, right hand or foot movement such as grasping and releasing hand. In this period, subject was also

instructed to stare a red dot during motor imagery to avoid eye movement artifacts. In the training session, to design a classifier that would be used in the testing session, we just collected the training trials for each motor imagery signal. At that time, the classifier had not been designed. Therefore, the yellow ball (feedback) was set to move into the target direction automatically.

In the online testing (feedback) phase, same experimental paradigm was used. However, the online feedback was provided in each trial. Thus, the yellow ball was controlled by the classified result of LDA which was analyzed from intention of each subject using the EEG data collected from 2 to 4 sec. We recorded 75 test trials for each class. One run consisted of 25 trials and we performed total three runs. Thus, in the one session experiment, total 60 offline and 75 online trials per class were collected for each subject. Both data were segmented from 2 to 4 sec after cue onset for further signal processing.

4.3 Preprocessing and Feature extraction

For preprocessing of experimental EEG dataset, we apply same procedures to all datasets and classification methods. First, we perform band pass filtering to eliminate the frequencies which are not related to motor imagery signals. In this study, we use fourth order Butterworth filter with 5 and 30 of cut off frequencies.

We use the CSP filtering and band power computation for feature extraction of MI EEG signals. For the SRC scheme, we use same procedure as explained in 2.4.3.

4.4 Adaptive SRC Schemes

To overcome inherent non-stationarity of EEG signals, we propose simple adaptive classification schemes based on the SRC method. In this study, we suggest two schemes, dictionary update method and incoherence based dictionary modification (IDM) method. Each scheme works with the conventional SRC method independently. In addition, both schemes can be incorporated as one combined adaptive SRC method. In the following subsections, we introduce each adaptive scheme.

4.4.1 Incoherence based Dictionary Modification Method

As we mentioned in Section 2.4.1, when a dictionary is incoherent a test signal from one particular class can be predominantly represented by the columns of the same class in the dictionary. The uncertainty principle (UP) [21][22] in the sparse representation theory dictates that a signal cannot be sparsely represented in both classes simultaneously. This phenomenon intensifies as the degree of incoherence of the dictionary increases.

An incoherent dictionary can be explained from the definition of mutual coherence of class-dictionary as shown in equation (1): $M(\mathbf{A}_L, \mathbf{A}_R) \triangleq \max\left\{ \left| \left\langle \mathbf{a}_{L,j}, \mathbf{a}_{R,k} \right\rangle \right| : j,k = 1,2,...,N_t \right\}$. In the SRC algorithm, we normalize the columns of dictionary **A**. Therefore, *M* measures the smallest angle between any pair of columns of two classes. When the value of *M* obtained from the two class-dictionaries is small, i.e., the cosine angle between two columns is large, we consider the dictionary incoherent. Due to the characteristics of the CSP filtering, i.e., CSP filters maximize the variance of the spatially filtered signal for one class data while minimizing it for the other class data, the CSP

features can be used for constructing incoherent dictionary (see Section 2). After applying CSP filtering, in the proposed IDM method, we aim to eliminate some training trials that have a high average cross coherence value with training trials of a different class. Thus, the eliminated training trials have features similar to those of many training trials of a different class. Therefore, we expect to further increase the incoherence of the dictionary by using the IDM method; this might lead to a high discrimination capability for training trials of two different classes.

In the IDM method, coherence value of the dictionary $\mathbf{A}$ can be simply estimated by each element of $\mathbf{G} =: \mathbf{A}^{\mathrm{T}}\mathbf{A}$. Thus, $\mathbf{G}(i, j)$ indicates the coherence value between i and j-th column of the dictionary. Therefore, $\mathbf{G}(i, j)$ is equal to $\mathbf{G}(j, i)$. For example, if the number of training trials of each class-dictionary is five, then the dimension of $\mathbf{G}$ is 10×10. From the $\mathbf{G}$, we focus on the *cross coherence* part between the two classes. Thus, we extract columns from 1-th to 5-th and rows from 6-th to 10-th of the $\mathbf{G}$ which are corresponding to the class 1 and class 2 respectively. Therefore, the dimension of cross coherence part is 5×5 in this example. We represent this cross coherence part as $\mathbf{G}_{\mathbf{CC}}$. Using the $\mathbf{G}_{\mathbf{CC}}$, we can easily check which trials of class 1 dictionary have large coherence values with trials from class 2 dictionary and vice versa.

Figure 4.2 shows example values of cross coherence $\mathbf{G}_{\mathbf{CC}} \in \mathbb{R}^{5\times5}$ and concept of the IDM method. In this figure, each number means the coherence value ranged from 1 to 9. Red colored elements represent high coherence values which are set up to be the values greater than or equal to 8. The values of last row and column represent the averaged value of five columns and rows respectively. In

this example, we set the number of elimination trials *n* equal to one. Thus, we aim to eliminate the highest average value for each column and row respectively.

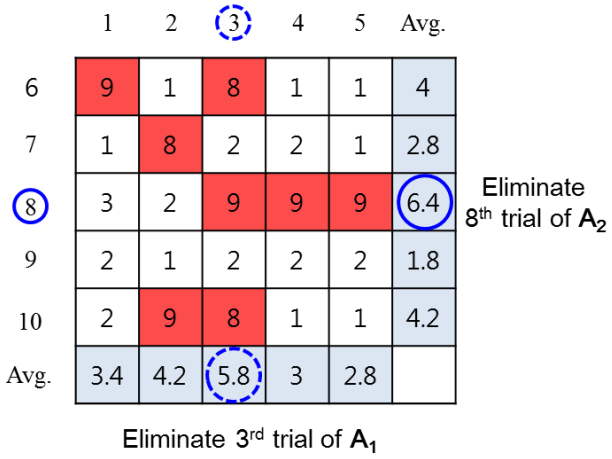


Figure 4.2 Example of incoherence based dictionary modification (IDM) method

From the averaged value of cross coherence, the third row and third column shows highest averaged value of 6.4 and 5.8. This means that 8-th row (8-th trial from class 2 dictionary) and third column (third trial from class 1 dictionary) shows high coherence value with many trials, i.e., many red colored elements, from the other class-dictionary. Therefore, we can eliminate the one trial in the each class-dictionary.

We summarize the incoherence based dictionary modification (IDM) algorithm as follows:

1. Set *n* the number of elimination trials
2. Compute the average value of each column of $\mathbf{G}_{CC}$
3. Collect the indices of column numbers which have n highest average coherence values
4. Eliminate n indices from original class-dictionary
5. Repeat 2~4 steps for row of $\mathbf{G}_{CC}$

For each subject dataset, we apply the IDM algorithm to the dictionary. After then, we perform the SRC steps with the modified dictionary.

4.4.2 Dictionary Update Methods

Normally, in motor imagery based BCI systems, a translation algorithm such as a classifier is designed using the collected training data. Then, an application device or program is controlled by using the classification algorithm in each test trial. However, because of the inherent non-stationarity of EEG, the classification performance deteriorates from the training to the test session in a BCI experiment. To overcome this drawback, many adaptive classification schemes are proposed. The main concept of the adaptive classification is re-adjustment (re-training) of the classifier for the new test data. On the other hand, in the SRC scheme, one important characteristic is that training (or parameter decision) of a classifier is not needed unlike in other decision hyper-plane based classification methods such as LDA and SVM. Thus, in the SRC scheme, a dictionary is simply formed by collecting the training feature vectors as columns of the dictionary. Then, using the dictionary sparse coding step is performed for each test data. Due to this unique classification mechanism, a simple intuitive method for adaptive SRC is dictionary update.

As we mentioned in Section 2.4.1, the dictionary $\mathbf{A}$ is formed by class-dictionary $\mathbf{A}_i = [\mathbf{a}_{i,1}, \mathbf{a}_{i,2}, ..., \mathbf{a}_{i,N_t}]$ in the SRC method. Each column vector $\mathbf{a}_{ij}$ is a *j*-th training feature vector of class *i*. Therefore, for each test trial in the online testing phase, a feature vector of a new test trial $\mathbf{y}$ can be easily updated as a new column of the dictionary. Then, characteristics of the test feature can

be applied into the dictionary while the online testing experiment is performed. And therefore, we can expect the classification performance of the online testing phase is not deteriorated.

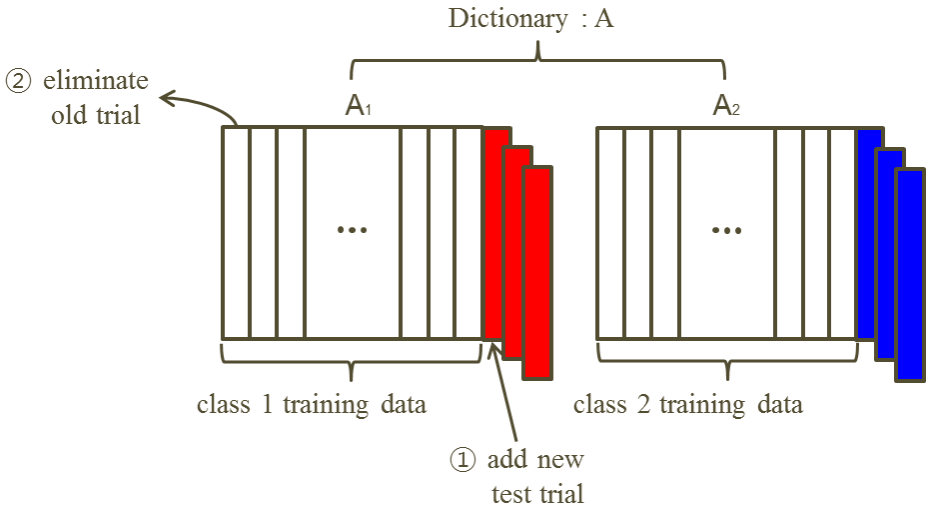


Figure 4.3 Concept of the proposed dictionary update rule

In this study, we consider four types of dictionary update rule, supervised accumulated update (SAU), supervised fixed update (SFU), unsupervised accumulated update (UAU) and unsupervised fixed update (UFU) rule. In our online experimental paradigm, as shown in Figure 4.1, a target class label is first provided as the position of the target bar. Then, subjects perform motor imagery corresponding to the class label information for each trial. In the supervised update rule, the target class label of test trials is used for updating the online test trials. Thus, a new test trial which has same class label of training trials in the class-dictionary is updated into the corresponding class-dictionary. However, this strategy is not practical for a general online scenario. Therefore, we also consider the unsupervised update rule. In the unsupervised update rule, class label information of the test trial is not used. Thus, each test trial is updated into the corresponding class-dictionary based on the

estimated result of the classifier, which is represented by the direction of the yellow ball movement shown in Figure 4.1.

For the case of accumulated update method, as shown in ① of Figure 4.3, all updated test trials are just stacked at the end (last column) of the class-dictionary based on the class label and classified result for SAU and UAU respectively. However, for the case of fixed update rule, SFU and UFU, the oldest training trial, i.e., the first training trial of the class-dictionary is eliminated as shown in ② of Figure 4.3 when each new test trial is updated. Note that if available training data in the dictionary is large enough and online testing phase is long, i.e., the number of test trials is large; the dictionary will be a fat matrix in the case of accumulated update rule. In this case, computation time for sparse representation is also increased. Therefore, in this study, we consider fixed update rule which has a same size dictionary, i.e., number of columns in the dictionary, with the original training dictionary. We compare computation time between accumulated and fixed update rule in Section 4.6.2.

4.5 Results

4.5.1 Evaluation Strategy

Using the online experimental dataset mentioned in Section 4.2, we aim to evaluate proposed adaptive SRC schemes, i.e., four dictionary update methods (supervised accumulated update (SAU), supervised fixed update (SFU), unsupervised accumulated update (UAU) and unsupervised fixed update (UFU) rule) and an incoherence based dictionary modification (IDM) method. From the multi session datasets of ten subjects, twelve session datasets are selected for evaluation of proposed

methods. In this selection, for a reliable assessment of classification methods, we choose datasets over 60% classification accuracy in the online experiment (In the binary classification, theoretical random chance level is 50%). Each session dataset consists of sixty training trials and seventy five test trials for each class.

In this Section 4, for the two class classification problems of the conventional SRC method, the dimension of the dictionary **A** is 10×120, i.e., m = 10 CSP features and N = 120 training trials. For each subject, 150 test trials where each has the same 10 dimension features are evaluated with dictionary **A**. For the proposed adaptive methods, we perform the incoherence based dictionary modification (IDM) method using the original dictionary **A**. After then, for each new test trial, we perform the each proposed dictionary update method for adaptation of test data.

Due to the inherent non-stationarity of EEG signals, online test data have different feature characteristics compared to training data. And therefore, even though classifier is well trained for training data, satisfactory classification performance is not guaranteed for online data. We expect that in the SRC method the proposed incoherence based dictionary modification (IDM) method is effective for proper dictionary design by maximizing incoherence between two classes. In addition, to overcome the non-stationarity of EEG, new test features will be applied into the original dictionary using updated new test trials from the proposed dictionary update method. Using online experimental dataset, we evaluate classification accuracy of the conventional SRC, each dictionary update method and IDM based adaptive SRC method. In addition, we also compare the classification results of the proposed methods with other adaptive classification methods such as adaptive LDA and SVM method.

4.5.2 Experimental results

To evaluate classification performance of the proposed adaptive SRC schemes, we compare classification accuracy (%) of proposed methods with that of conventional SRC method using the online experimental dataset of twelve motor imagery sessions. Table 4.1 shows the classification accuracy of the SRC and the proposed dictionary update based SRC methods with and without IDM method. For fair comparison, we set the same value of n (the number of elimination trials of IDM) of 10 for all subjects and all IDM based adaptive SRC methods.

Table 4.1 Classification accuracy of conventional SRC and proposed adaptive SRC schemes (SRC_SAU, SRC_SFU, and SRC_USU) for 12 session datasets. We present the classification accuracy (%) of each method with and without IDM. The highest classification accuracy for each dataset is highlighted in bold.

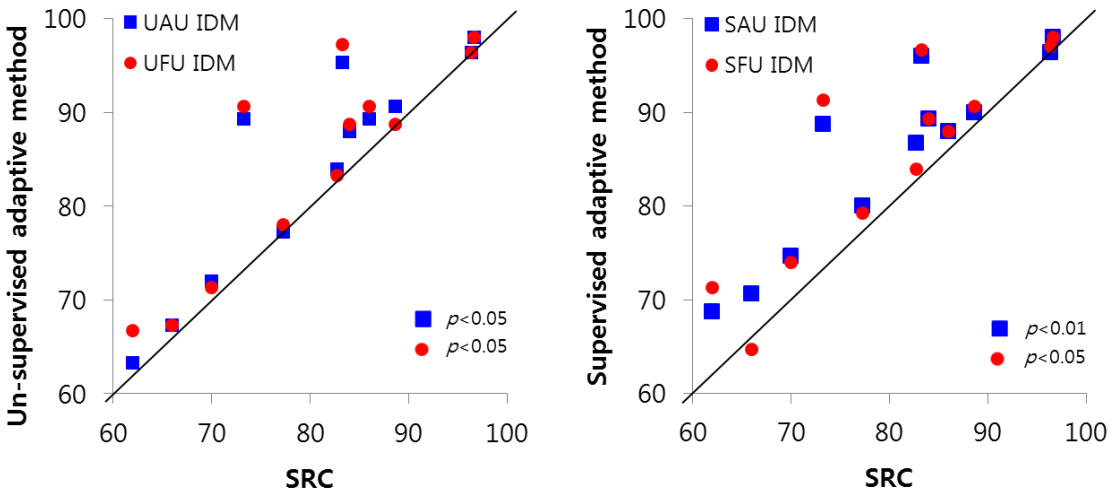
Data	SRC		SRC_SAU		SRC_SFU		SRC_UAU		SRC_UFU	
	w/o IDM	w/ IDM	w/o IDM	w/ IDM	w/o IDM	w/ IDM	w/o IDM	w/ IDM	w/o IDM	w/ IDM
1	66	66.7	67.3	**70.7**	66.0	64.7	66.0	67.3	66.0	67.3
2	86	86.7	88.0	88.0	88.0	88.0	87.3	89.3	82.7	**90.7**
3	88.7	**90.7**	90.0	90.0	89.3	**90.7**	90.0	**90.7**	90.7	88.7
4	96.4	96.4	96.4	96.4	**97.1**	**97.1**	96.4	96.4	96.4	96.4
5	83.3	89.3	93.3	96.0	96.0	96.7	93.3	95.3	94.7	**97.3**
6	82.7	78.7	**86.7**	**86.7**	84.0	84.0	80.0	84.0	80.7	83.3
7	77.3	75.3	78.0	**80.0**	78.7	79.3	76.7	77.3	79.3	78.0
8	73.3	88.0	88.7	88.7	89.3	**91.3**	78.0	89.3	84.7	90.7
9	70.0	**75.3**	74.0	74.7	73.3	74.0	70.0	72.0	70.0	71.3
10	62.0	64.0	66.0	68.7	67.3	**71.3**	62.0	63.3	68.0	66.7
11	84.0	87.3	88.7	**89.3**	88.7	**89.3**	86.7	88.0	88.0	88.7
12	96.7	96.0	97.3	**98.0**	97.3	**98.0**	96.7	**98.0**	96.7	**98.0**
Mean	80.5	82.9	84.5	**85.6**	84.6	85.4	81.9	84.3	83.1	84.8
Std.	11.13	10.74	10.69	9.94	10.99	10.89	11.73	11.64	10.84	11.40

From the results of Table 4.1, all five methods with IDM show better mean classification accuracy than the without IDM method. Thus, the proposed IDM method is effective for the SRC framework. Furthermore, the proposed simple dictionary update methods with and without IDM show improved mean classification accuracy than the conventional SRC method. Supervised update methods, i.e., SAU and SFU, show more improved results than the unsupervised methods, UAU and UFU. However, mean difference between SAU/ SFU with IDM and UAU/ UFU with IDM is not much.

For further analysis, in Figure 4.4, we investigate the comparison of the classification accuracy of twelve datasets using scatter plots. Each point indicates the classification accuracy of each dataset which is used for computing mean classification accuracy in Table 4.1. Figure 4.4 (a) shows the comparison results between the SRC and the two supervised dictionary update methods with IDM. Classification accuracies of the SRC and supervised methods are represented in X and Y-axis respectively. For the supervised methods (Y-axis), blue square points indicate the SAU with IDM method and red circle points indicate the SFU with IDM method. Similarly, Figure 4.4 (b) shows the comparison results between the SRC and the two unsupervised dictionary update methods.

From the results of Figure 4.4 (a), both SAU and SFU with IDM show higher classification accuracies than the SRC method for eleven datasets. Thus, the eleven data points positioned over the black linear-line which indicates the same classification accuracy between SRC and proposed methods. On the Figure 4.4 (b), we also observe that the both UAU and UFU IDM show higher

classification accuracies than the SRC for ten datasets. In addition, p-values obtained from paired t-test are smaller than 0.05 for all comparisons between the SRC and proposed methods in Figure 4.4.



(a) (b)

Figure 4.4 Comparison of classification accuracy of all twelve datasets (a): Scatter plot of classification accuracies between conventional SRC (X-axis) and the both supervised update methods SAU and SFU with IDM (Y-axis) (b): Scatter plot of classification accuracies between conventional SRC (X-axis) and the both unsupervised update methods UAU and UFU with IDM (Y-axis).

To evaluate the effect of the proposed methods, we analyze one dataset in the feature space. Figure 4.5 shows scatter plots of training and test features of dataset 5 used in Table 4.1. For ease of visualization, we use two-dimensional feature spaces which are corresponding to the first and the last CSP filters. In Figure 4.5, the red and black x marks indicate the 60 training and 75 test features for one class, respectively. On the other hand, the blue and green circles indicate the 60 training and 75 test features for another class, respectively. Each class training and test data element is fitted by a Gaussian distribution. Therefore, we can easily check the distribution change from the training to the test data during the experimental sessions. When the distribution of the test data is changed from that

of the training data, the previously designed dictionary based on the training data is not optimal for the classification of new test data.

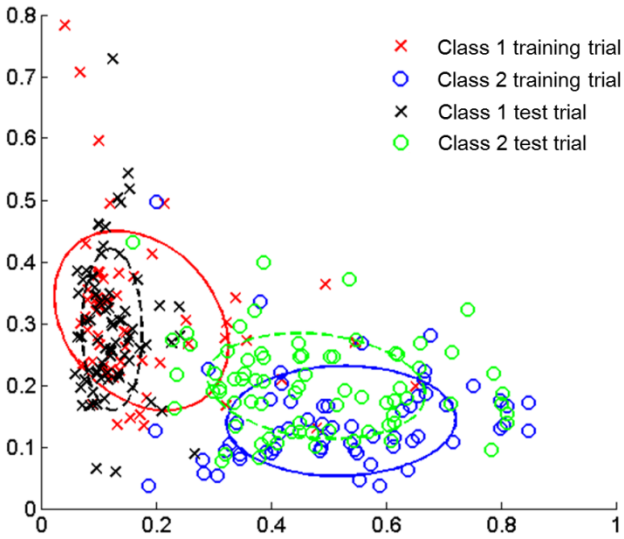
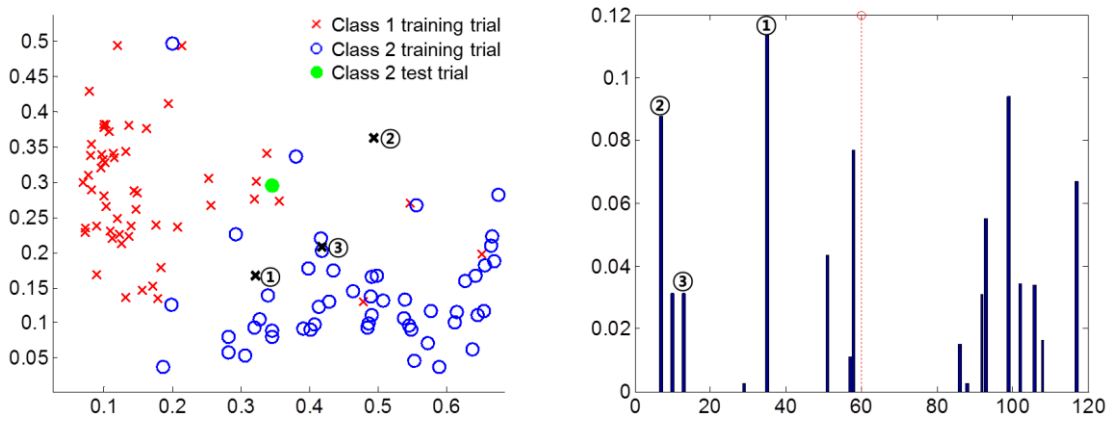


Figure 4.5 Scatter plot of training and test features for two different classes in two dimensional feature spaces using an example dataset 5. All training and test samples are scattered and fitted by Gaussian distribution for illustration.

Figure 4.6 shows one classification instance of a test trial, which is represented by a filled green point (class 2) in the left figure. In this test, the test feature is not correctly classified, i.e., classified as class 1, by the conventional SRC without IDM method. All training features in the dictionary of classes 1 and 2 shown in Figure 4.5 are utilized for the classification of the test feature without the use of any adaptation techniques. Figure 4.6 (a) shows the coefficients recovered by the conventional SRC for the test feature represented in the left figure. The X-axis represents the training trial number (column number) of the dictionary, and the red dotted line denotes the boundary of two different classes. In the Figure 4.6 (b), the numbering ①, ② and ③ represent the coefficients corresponding to the training trials of black x marks ①, ② and ③ in the left figure. Because the three training points of

class 1 are used for the sparse representation of the test trial and have large coefficient values, the test feature is classified as class 1 by using the minimum residual rule in equation (2).

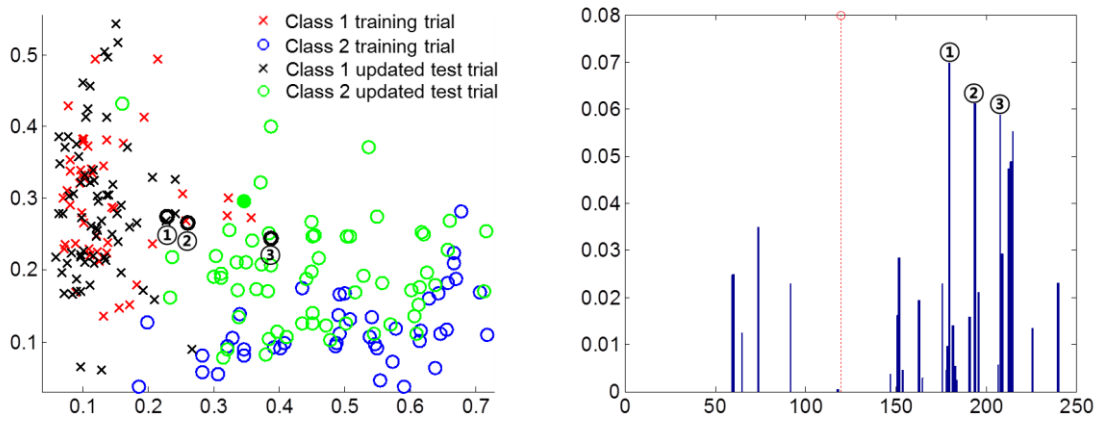


(a) (b)

Figure 4.6 Classification results of conventional SRC for one test sample of dataset 5 for. (a): Scatter plot of training features for two classes and one test feature of class 2. (b): Sparse representation results of one test feature shown in left figure from the conventional SRC. X-axis represents the training trial number in dictionary and Red dotted line means the boundary of two different classes.

On the other hand, Figure 4.7 shows the classification results of SRC_UAU IDM for the same test trial used in Figure 4.6. In Figure 4.7 (a), we can see that some training features which are originally positioned at the area of different class features including the black x marks ①, ② and ③ in Figure 4.6 (a) are effectively eliminated by the IDM method. In addition, new test trials represented by the black x marks and the green and black circles are also updated before the classification of the current test trial, which is represented by the filled green circles. From the result of Figure 7 (b), we conclude that the test trial is correctly classified as class 2 and the three updated test trials represented by black circles ①, ② and ③ in the left figure have large coefficients. Therefore, for the classification of new test trials, IDM and the dictionary update method in SRC are very effective, and we can see that the

proposed methods with IDM show relatively improved classification accuracy compared to the conventional SRC from the results of dataset 5, presented in Table 4.1.



(a) (b)

Figure 4.7 Classification results of SRC_UAU IDM for the same test sample in Figure 6. (a): Scatter plot of training features for two classes and one test feature of class 2. (b): Sparse representation results of one test feature shown in left figure from the SRC_UAU IDM.

Next, we compare the classification accuracy of the conventional SRC and the proposed adaptive SRC methods with the non-adaptive and adaptive LDA and SVM classification methods using our experimental dataset in Table 4.2. The LDA and SVM are widely used classification methods in many EEG based BCI researches. For the adaptive LDA and SVM methods, first, linear decision hyper-plane is chosen from training data. Then in the testing session, the decision hyper-plane is re-trained for new test sample. We only consider supervised adaptation for the LDA and SVM methods.

From the results presented in Table 4.2, we can first see that the conventional SRC exhibits better mean classification accuracy than the non-adaptive LDA and SVM methods. These results are consistent with those of the previous results in Section 2 and 3. Second, the proposed adaptive SRC

methods show better mean classification accuracy than the other adaptive LDA and SVM methods. Note that even though the accuracy difference between the unsupervised adaptive SRC methods and adaptive SVM method is not much, in the conventional adaptive methods, re-training (re-adjustment) of the decision hyper-plane for new test data is time consuming process. However, in the proposed methods, dictionary update for adaptation of each test sample is very simple process and re-training of classifier is not needed.

Table 4.2 Comparison of classification accuracy (%) between conventional non-adaptive classification methods (LDA, SVM, and SRC) and adaptive classification methods (including the proposed adaptive SRC schemes). The highest classification accuracy for each dataset is highlighted in bold.

Data	LDA	Adap. LDA	SVM	Adap. SVM	SRC	SRC _SAU IDM	SRC _SFU IDM	SRC _UAU IDM	SRC _UFU IDM
1	56.0	62.7	68.7	69.3	66.0	**70.7**	64.7	67.3	67.3
2	88.0	87.3	88.0	88.0	86.0	88.0	88.0	89.3	**90.7**
3	87.3	86.7	86.0	86.0	88.7	90.0	**90.7**	**90.7**	88.7
4	94.3	94.3	95.7	95.0	96.4	96.4	**97.1**	96.4	96.4
5	78.0	84.0	80.0	89.3	83.3	96.0	96.7	95.3	**97.3**
6	79.3	82.0	84.7	**90.7**	82.7	86.7	84.0	84.0	83.3
7	68.7	74.0	71.3	**80.0**	77.3	**80.0**	79.3	77.3	78.0
8	84.7	89.3	70.7	89.3	73.3	88.7	**91.3**	89.3	90.7
9	70.7	74.0	69.3	73.3	70.0	**74.7**	74.0	72.0	71.3
10	53.3	63.3	58.0	62.7	62.0	68.7	**71.3**	63.3	66.7
11	79.3	82.7	70.0	87.3	84.0	**89.3**	**89.3**	88.0	88.7
12	87.3	91.3	94.0	95.3	96.7	**98.0**	**98.0**	**98.0**	**98.0**
Mean	77.2	81	78	83.9	80.5	**85.6**	85.4	84.3	84.8
Std.	12.84	10.36	11.70	10.36	11.13	9.94	10.89	11.64	11.40

4.6 Discussions

4.6.1 Results for Public Dataset

For the evaluation of the proposed methods, we use a public dataset obtained from Dataset IVc of BCI Competition III [47]. In this dataset, the test data were separately recorded for more than 3 hours after the acquisition of the training data. Therefore, the distribution of some EEG features could be effected by non-stationarities. This dataset was recorded from a healthy subject. He sat in a comfortable chair with his arms resting on the armrests. The training dataset consists of the data of the first three (non-feedback) sessions. In all, 210 training trials (105 for each class) were obtained. The visual cues (letter presentation) indicated for 3.5 sec which of the following two motor imageries that the subject had to perform: (L) left hand and (F) right foot. The target cues were presented at intervals of random length ranging from 1.75 to 2.25 sec, in which the subject could relax. In the test sessions, total 280 test trials (140 for each class) were recorded. The experimental setup was similar to the setup of the training sessions, but the motor imagery had to be performed for 1 sec only, compared to 3.5 sec in the training sessions. The recording was made using BrainAmp amplifiers and a 128-channel Ag/AgCl electrode cap from ECI. A total of 118 EEG channels were measured at the positions of the extended international 10/20 system. Signals were band-pass filtered between 0.05 and 200 Hz, and then digitized at 1000 Hz.

Table 4.3 shows the classification accuracy of the public dataset for conventional SRC and the four proposed adaptive SRC schemes when the number of elimination trials n is varied from 0 (no IDM) to

30. For this dataset, six CSP filters are used for feature extraction, and thus, the dimension of dictionary **A** is 6×210 for the original SRC. In all, 280 test trials are classified by each classification method. From the results presented in Table 4.3, we find that all proposed adaptive SRC methods exhibit improved classification accuracy compared to the conventional SRC method irrespective of the value *n* of IDM. Supervised dictionary update methods (SAU and SFU IDM) show better classification accuracy than the unsupervised methods (UAU and UFU IDM); however, the difference is very small (within 1%). Further, the difference between the accumulated (SAU and UAU IDM) and the fixed dictionary update methods (SFU and UFU IDM) is more small and negligible for this dataset.

Table 4.3 Classification accuracy (%) of conventional SRC and the proposed adaptive SRC methods for the public BCI competition dataset.

n of IDM	SRC	SRC_SAU IDM	SRC_SFU IDM	SRC_UAU IDM	SRC_UFU IDM
0	92.5	**95.36**	**95.36**	93.93	94.64
5	92.86	**96.07**	95.71	94.64	94.64
10	90	95.36	**95.71**	93.93	93.93
15	92.86	**95.36**	**95.36**	94.64	94.64
20	91.43	95.36	**95.71**	95.36	94.64
30	91.79	**95**	**95**	94.64	94.64
Mean	91.91	95.42	95.48	94.52	94.52

4.6.2 Comparison between Proposed Adaptive Schemes

In this section, first, we compare the accumulated and fixed dictionary update rule for each supervised and unsupervised dictionary update method. From the results of Table 4.1, the mean difference between SRC_SAU and SRC_SFU with IDM is just 0.2%. For the unsupervised case,

SRC_UAU and SRC_UFU with IDM exhibit a mean difference of 0.5%. To analyze the statistical significance of the mean differences, we perform the paired *t*-test for the accuracy of each subject. The obtained *p*-values of the *t*-test are larger than 0.05 for the comparisons of the accumulated and the fixed update rule, which means that the differences are not statistically significant. As we mentioned in Section 4.4.2, when the number of original training trials in the dictionary and that of the updated new test trials are large, the computation time of the accumulated dictionary update based SRC method might be increased to solve the sparse coding step, i.e., equation (5), by using L1 minimization as compared to the fixed dictionary update based SRC method. Thus, in the fixed update rule, the dictionary size is fixed for all test trials and the computation time for sparse coding is not increased. However, in the accumulated update rule, the dictionary size is increased in every test trial, and therefore, the computation time for the sparse coding step is also increased.

We compare the running time (computation time) of the accumulated and fixed dictionary update methods. Because of the number of training trials and that of the test trials of the competition dataset, which is used in Section 4.6.1 (210 and 280), are larger than our dataset (120 and 150), we use the competition dataset to evaluate the running time. The *tic* and *toc* MATLAB commands are used for measuring the running time of the sparse coding step in the SRC algorithm. We repeat 100 times and measure the average running time for each method. For a single test trial, the average running time of the sparse coding step in SRC_SAU and SFU are 5.47 msec and 4.29 msec respectively. Further, the SRC_UAU and UFU show the average running time of 5.45 msec and 4.26 msec for the sparse

coding step, respectively. Therefore, for a single test trial, the differences in the running time between the accumulated and the fixed update rule are very small and negligible for online BCI applications.

Second, we investigate supervised and unsupervised dictionary update methods. From the results presented in Table 4.1, we find that the mean difference between SRC_SAU and SRC_UAU with IDM is 1.3%. For this comparison, we obtained a *p*-value of 0.04 from the paired *t*-test. For the unsupervised case, the mean difference between SRC_SFU and SRC_UFU with IDM is 0.6% and the obtained *p*-value is larger than 0.05. Even though the mean differences are not much, all supervised methods consistently show better mean classification accuracy than the unsupervised methods for our dataset and the public dataset presented in Tables 4.1 and 4.3, respectively. In the unsupervised dictionary update method, the class labels of the test trials are determined by the results of the classifier. Unfortunately, the classifier usually does not provide perfect classification results for all test trials because of the non-stationarity of EEG. Few incorrectly classified test trials are also updated in a different class-dictionary with the original target class. These trials affect the sparse coding step in the SRC algorithm. In addition, in our online dataset, LDA classifier is used for providing feedback results. Therefore, user adaptation from SRC cannot be obtained in the unsupervised method. These might be the reasons that the unsupervised methods exhibit lower mean classification accuracy than the supervised methods in this study. However, from the results for our dataset and the public dataset, we find that the unsupervised methods still show improved classification results compared to the original SRC.

4.6.3 Analysis of IDM Method

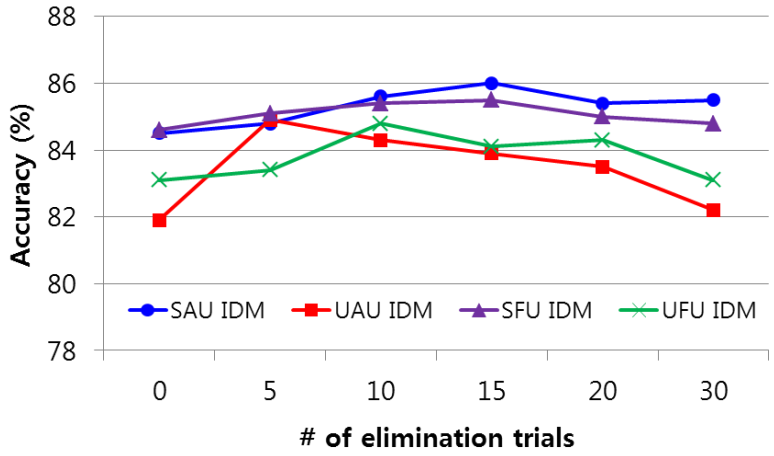


Figure 4.8 Average classification accuracy of SAU IDM and UAU IDM when the number of elimination trials *n* is varied.

As shown in the results of Table 4.3, the classification accuracy of IDM based SRC methods may vary on the basis of the value *n* of IDM. The value *n* can be heuristically chosen to optimize the classification accuracy. In this section, we analyze the effect of the number of elimination trials *n* of IDM by using our experimental dataset. In the results presented in Table 4.1, for a fair comparison, we set the same value of *n* of 10 for all twelve datasets. For the same datasets, in Figure 4.8, we compute average classification accuracy over all datasets when the number of elimination trials of SAU, SFU, UAU and UFU IDM is varied from 0 to 30. From the results of Figure 4.8, the optimal number *n* is different for each method. This means that there is a place to improve classification performance of IDM based adaptive SRC method by finding optimal *n* for each method and also each subject dataset. In Figure 4.8, compared to the results of supervised update methods average accuracy is decreased with the large value of *n* in the case of unsupervised update methods. This might be

because if the number of elimination trials n is large, number of training trials is decreased in the dictionary. Thus, the role for classification task of updated new test trials is increased. However, in the case of unsupervised method, class label of new test trials is not always correctly updated. Therefore, for the unsupervised update methods with IDM, the value n is needed to choose more carefully.

Next, we analyze the effect of the incoherence based dictionary modification (IDM) method. As we mentioned in Section 4.4.1, we propose an IDM method to make more incoherent dictionary after applying the CSP filtering. Incoherence of dictionary can be measured by mutual coherence value M introduced in equation (1). To evaluate the change in the coherence value, we measure the M value of SRC without IDM and with IDM method. From the average results over twelve datasets, The SRC without IDM shows 0.983 value of M. On the other hand, the SRC with IDM shows 0.934 value of M. This means that after applying the IDM method, we can make more incoherent dictionary than the without IDM method.

4.7 Summary

Because of the inherent non-stationarity of EEG signals, performance degradation is an inevitable phenomenon in EEG based BCI systems. In particular, an already designed classifier by the training data does not guarantee satisfactory classification accuracy for new test data in the online feedback stage. In this paper, we propose dictionary update methods with incoherence based dictionary modification (IDM) as adaptive SRC schemes to compensate for the non-stationary effects. We

consider supervised/unsupervised and accumulated/fixed dictionary update rules with IDM. With the unique classification mechanism of the SRC, i.e., a fixed decision rule is not required for the classification, in the proposed dictionary update methods, the test data are easily updated and utilized for the classification of other new test data without requiring any additional computation. In addition, in the IDM algorithm, we try to create a maximally incoherent dictionary for SRC by using a simple incoherence measure of the training data. By using two online motor imagery based BCI experimental datasets, we evaluate the classification performance of the proposed adaptive schemes. From the results, we find that the proposed IDM based adaptive SRC schemes show improved classification results compared to the conventional SRC. Further, unsupervised adaptive SRC schemes that are more practically applicable in BCI exhibit competitive classification accuracy than other adaptive LDA and SVM methods.

References

[1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, T. M. Vaughan, "Brain-computer interfaces for communication and control", *Clin.Neurophysiol.* vol. 113, pp. 767-791, 2002.

[2] J. R. Wolpaw, D. J. McFarland, G. W. Neat and C. A. Forneris, "An EEG-based brain-computer interface for cursor control", *Electroencephalogr. Clin. Neurophysiol.* vol. **78,** pp. 252–259, 1991.

[3] J. R. Wolpaw, D. J. McFarland, "Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans," PNAS, vol.101, no.51, pp.17849-17854, December 2004.

[4] G. Dornhege, J. R. Millán, T. Hinterberger, D. J. McFarland, K. R. Müller, "Toward Brain-Computer Interfacing", *The MIT Press*, pp. 213–215, 2007.

[5] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, K. R. Müller, "Optimizing spatial filters for robust EEG single-trial analysis" *IEEE Signal Process Mag.*, vol. 25 no. 1, pp. 41–56, 2008.

[6] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, B. Arnaldi, "A review of classification algorithms for EEG-based brain-computer interfaces", *J. Neural Eng.* vol. 4 no. 2, R1–R13, 2007.

[7] G. Pfurtscheller, D. Flotzinger, and J. Kalcher, "Brain-computer interface-a new communication device for handicapped persons," J. Microcomput. Appl., vol. 16, pp. 293-299, 1993.

[8] D. Donoho, "Compressed sensing", *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289–1306, 2006.

[9] R. Baraniuk, “Compressive sensing” *IEEE Signal Process. Mag.*, vol. 24 np. 4, pp. 118–121, 2007.

[10] K. Huang and S. Aviyente, “Sparse representation for signal classification”, *Adv. Neural Inf. Process. Syst.*, vol. 19, pp. 609–616, 2006.

[11] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry and Y. Ma, “Robust face recognition via sparse representation”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31 no. 2, pp. 210–227, 2009.

[12] M. Yang and L. Zhang, “Gabor Feature based Sparse Representation for Face Recognition with Gabor Occlusion Dictionary”, *In ECCV*, 2010.

[13] J. F. Gemmeke, T. Virtanen and A. Hurmalainen, “Exemplar-based sparse representations for noise robust automatic speech recognition”, *IEEE Trans. Audio, Speech, Lang. Proc.* vol. 19 no. 7, pp. 2067–2080, 2011.

[14] Y. Shin, S. Lee, J. Lee and H.-N. Lee, “Sparse representation-based classification scheme for motor imagery-based brain-computer interface systems”, *Journal of Neural Engineering*, no. 9 056002, 2002.

[15] Y. Shin, S. Lee, M. Ahn, H. Cho, S. C. Jun and H.-N. Lee, “Noise Robustness Analysis of Sparse Representation based Classification Method for Non-stationary EEG Signal Classification”, *Biomedical Signal Processing and Control* vol. 21, pp. 8-18, 2015.

[16] Y. Shin, S. Lee, M. Ahn, H. Cho, S. C. Jun and H.-N. Lee, “Simple Adaptive Sparse Representation based Classification Schemes for EEG based Brain-Computer Interface Applications”, *Computers in Biology and Medicine*, vol. 66, pp. 29-38, 2015.

[17] P. L. Nunez, R. Srinivasan, A. F. Westdorp, R. S. Wijesinghe, D. M. Tucker, R. B. Silberstein and P. J. Cadusch, “EEG coherency I: statistics, reference electrode, volume

conduction, Laplacians, cortical imaging, and interpretation at multiple scales", *Electroencephalogr. Clin. Neurophysiol.*, vol. 103, pp. 499–515, 1997.

[18] B. Graimann, B. Allison and G. Pfurtscheller, "Brain-Computer Interfaces: Revolutionizing Human-Computer Interaction", *Springer*, 2010.

[19] B. Blankertz, Berlin Brain-Computer Interface, http://www.bbci.de/

[20] G. Pfurtscheller and C. Neuper, "Motor imagery and direct brain–computer communication", *Proc. IEEE*, vol. 89, pp. 1123–1134, 2001.

[21] D. L. Donoho and X. Huo, "Uncertainty principles and ideal atomic decomposition", *IEEE Trans. Inf. Theory*, vol. 47, pp. 2845–2862, 2001.

[22] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ1 minimization", *Proc. Natl. Acad. Sci.* vol. 100, no. 5, pp. 2197–2202, 2003.

[23] E. Candès E, J. Romberg and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements", *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[24] S. Chen, D. L. Donoho and M. Saunders, "Atomic decomposition by basis pursuit" *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[25] D. L. Donoho, V. Stodden and Y. Tsaig, SparseLab, http://sparselab.stanford.edu/

[26] V. Bostanov, "BCI competition 2003–data sets ib and iib: feature extraction from event-related brain potentials with the continuous wavelet transform and the t-value scalogram", *IEEE Trans. Biomed. Eng.* vol. **51,** pp. 1057–1061, 2004.

[27] L. Wasserman, "All of Statistics: A Concise Course in Statistical Inference", *Springer*, pp. 63–64, 2010.

[28] P. Shenoy, M. Krauledat, B. Blankertz, R. P. N. Rao and K. R. Müller, "Towards adaptive classification for BCI", *J. Neural Eng.*, vol. 3, pp. R13–R23, 2006.

[29] W. Samek, C. Vidaurre, K. R. Müller and M. Kawanabe, "Stationary common spatial patterns for brain-computer interfacing", *J. Neural Eng.*, vol. 9, no. 2, 026013, 2012.

[30] M. Kawanabe, W. Samek, K. R. Müller and C. Vidaurre, "Robust common spatial filters with a maxmin approach", *Neural Comput.*, vol. 26, no. 2, pp. 1-28, 2014.

[31] M. A. Oskoei, J. Q. Gan and H. Huosheng, "Adaptive schemes applied to online SVM for BCI data classification", *Inter. Conf. IEEE on Eng. in Medicine and Biology Society (EMBC)*, pp. 2600-2603, 2009.

[32] C. Vidaurre, M. Kawanabe, P. von Bünau, B. Blankertz and K. R. Müller, "Toward unsupervised adaptation of LDA for brain–computer interfaces", *IEEE Trans. Biomed. Eng.* vol. 58, pp. 587–597, 2011.

[33] A. Schlögl, F. Lee, H. Bischof and G. Pfurtscheller, "Characterization of four-class motor imagery EEG data for the BCI-competition", *J. Neural Eng.*, vol. 2, no. 4, pp. L14–L22, 2005.

[34] MathWorks: http://www.mathworks.co.kr/kr/help/stats/support-vector-machines-svm.html

[35] C. W. Hsu, C. C. Chang and C. J. Lin, "A practical guide to support vector classification", *Tech. rep., Department of Computer Science, National Taiwan University*, 2003.

[36] H. Morioka, A. Kanemura, J. I. Hirayama, M. Shikauchi, T. Ogawa, S. Ikeda, M. Kawanabe and S. Ishii, "Learning a common dictionary for subject-transfer decoding with resting calibration", *NeuroImage*, vol. 111, pp. 167-178, 2015.

[37] A. Kübler, N. Neumann, B. Wilhelm, T. Hinterberger and N. Birbaumer, "Predictability of brain–computer communication", *J Psychophysiol*., vol. 18, pp. 121–129, 2004.

[38] E. W. Sellers, A. Kübler and E. Donchin, “Brain-computer interface research at the University of South Florida Cognitive Psychophysiology Laboratory: the P300 Speller”, *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no.2, pp. 221–224, 2006.

[39] A. Y. Yang, Z. Zhou, A. Ganesh, S. S. Sastry, Y. Ma, “Fast l1-minimization algorithms for robust face recognition”, *IEEE Trans. Image process.*, vol. 22, no. 8, pp. 3234–3246, 2013.

[40] http://emotiv.com/store/epoc-detail/

[41] http://www.quasarusa.com/products_dsi.htm

[42] Y. M. Chi, Y. T. Wang, Y. Wang, C. Maier, T. P. Jung and G. Cauwenberghs, “Dry and noncontact EEG sensors for mobile brain-computer interfaces”, *IEEE Trans Neural Syst. Rehabil. Eng.*, vol. 20, no. 2, pp. 228–235, 2012.

[43] L. D. Liao, C. Y. Chen, I. J. Wang, S. F. Chen, S. Y. Li, B. W. Chen, J. Y. Chang and C. T. Lin, “Gaming control using a wearable and wireless EEG-based brain-computer interface device with novel dry foam-based sensors”, *J. Neuroeng. Rehabil.*, vol. 9, pp. 5. DOI:10.1186/1743-0003-9-5, 2012.

[44] C. T. Lin, B. S. Lin, F. C. Lin and C. J. Chang, “Brain computer interface-based smart living environmental auto-adjustment control system in UPnP home networking”, *IEEE Syst. J.* vol. 8, pp. 363–370, 2014.

[45] J. D. R. Millán, "On the need for on-line learning in brain-computer interfaces", *Proc. Int. Joint Conference Neural Networks*, pp.2877 -2882, 2004.

[46] Y. Li and C. Guan, “An extended EM algorithm for joint feature extraction and classification in brain-computer interfaces”, *Neural Comput.*, vol. 18, no. 11, pp. 2730-2761, 2006.

[47] http://www.bbci.de/competition/iii/desc_IVc.html

Acknowledgement

2009년 가을 처음 광주과학기술원에서 석사과정을 시작하여 2016년 2월까지 약 6년반 동안의 시간을 보내고 이제 박사과정을 마무리 하고자 합니다. 돌아보면 이곳에 와서 연구를 할 수 있었던 것은 내 인생에 정말 큰 기회이자 소중한 경험이었던 것 같습니다. 박사과정을 마무리하는 시점이 왔지만 현재의 저는 정말 많이 부족하고 평생 배우고 노력하며 살아가야겠다는 것을 느끼게 됩니다. 동시에 열심히 한다면 불가능 한 것도 없다는 것을 느끼게 됩니다.

제가 박사학위를 받을 수 있기까지 너무도 많은 분들께서 도와주셨습니다. 그 분들이 없었다면 아마 박사학위를 못 받았거나 훨씬 오랜 시간이 걸렸을 것이라 생각합니다.

먼저 막내아들에게 언제나 물심양면으로 지원을 해주시는 아버지 어머니, 두 분이 계시 기에 제가 편안한 마음으로 연구에 전념할 수 있었습니다. 또한 이곳에서 가까운 목포에 부모님께서 살고 계셔서 힘들고 외로운 시간에 같이 시간을 보내며 의지하고 이겨낼 수 있었습니다. 다시 한번 부모님의 지원과 무한한 사랑에 감사합니다. 평생 효도하며 살겠습니다.

박사학위 과정 중에 만나 결혼까지 하게 된 나의 평생의 동반자이자 사랑스런 부인 화진이에게도 너무나 감사하고 사랑한다는 말을 전하고 싶습니다. 정말 힘들었던 시절에 저하나만 바라보고 저의 옆에서 너무도 큰 사랑과 지원을 해주었습니다. 앞으로도 평생 함께하며 행복하고 건강하게 살자고 말하고 싶습니다.

또한, 제가 처음 GIST에 입학하였을 때부터 저를 받아주시고 박사졸업까지 지도해주신 지도교수 이흥노 교수님께도 진심으로 감사의 마음을 전합니다. 논문 지도뿐만 아니라 제 결혼식의 주례를 맡아주시어 인생의 스승으로써 여러 가르침을 주신 점 평생 잊지 않겠습니다. 이 밖에도, 박사학위 논문 심사위원을 맡아주시고 여러 조언도 많이 해주신 전성찬 교수님, 이보름 교수님, 김소희 교수님 그리고 Benjamin Blankertz 에게도 진심으로 감사드립니다.

오랜 시간을 한 연구실에서 보냈던 우리 INFONET 연구실 학생들에게도 감사의 마음을 전하고 싶습니다. 가장먼저 저에게 정말 많은 도움을 주고 같이 보낸 추억이 너무도 많은 저의 파트너 승찬이에게 너무도 고맙습니다. 승찬이와 떨어져 지낼 생각에 가슴이 아픕니다. 평생 연락하며 가까이 지낼 수 있는 형 동생으로 남고 싶다고 전하고 싶습니다. 그리고 동기로 같이 입학하여 항상 제가 물어보는 질문에 잘 대답해 주던 상준이와 듬직하게 연구실을 이끌어 주시는 웅비형, 또한 항상 잘 따라주고 믿음직한 후배 주성이, 해웅이, 승윤이, 재혁이, 들어 온지 얼마 안된 대영이, 재원이, 철순이까지 모두 같이 연구실 생활을 할 수 있어서 행복했다고 전하고 싶습니다. 항상 영어나 연구 관련해서 친절하게 알려주었던 외국인 친구들인 Pavel, Zafar, Asif, Evgenii, Nitin 에게도 감사하고 새로 온지 얼마 안된 Yaseen 과 Rohit에게도 앞으로 GIST 에서 연구활동 건강하게 잘 하라고 전하고 싶습니다.

이곳에서 같이 생활했었던 맏형 진택이형, 환철이형, Oliver, 변상선교수님, 정민이, 수길이, 재건이, 형원이, 준호, 현주씨, 수정씨 등 이곳에서 맺었던 모든 인연들에게 같이 시간을 보내서 즐거웠고 늘 건강 하라고 전하고 싶습니다. 특별히, 다른 연구실이었지만

같은 주제를 연구하며 워크샵도 같이하고 토의도 많이 했던 민규형, 호현이, 상태, 동현이 에게도 감사의 마음을 전합니다.

마지막으로 사랑하는 가족 할머니, 형, 누나, 형수님, 매형, 우리 조카 지환이 그리고 장인 장모님을 포함한 처갓집 식구들까지 모두 제가 학생이라는 이유로 가족 노릇을 제대로 못한 것 같아서 죄송스럽고 앞으로 가족들에게 더 잘하도록 노력하며 살겠습니다.

지금까지 인생을 살아오며 느낀 점은 사람이 살아가면서 갖게 되는 욕심도 걱정도 끝이 없다는 것 입니다. 그러기에 지금 현재에 행복을 느낄 줄 알며 현재에 충실하게 살아가는 것이 인생을 잘사는 방법이 아닐까 생각이 듭니다. 앞으로 제 인생에 어떤 그림들이 펼쳐질지 모르지만 항상 현재에 만족할 줄 알고 하루하루를 충실히 살아가는 사람이 되도록 노력하겠습니다.