# Feature Extraction, Feature Selection and Dimensionality Reduction Techniques for Brain Computer Interfaces

By
Tian Lan

A thesis submitted to
the Department of Biomedical Engineering of
the Oregon Health & Science University
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
in
Electrical Engineering

July 2011

The Dissertation “Feature extraction, feature selection and dimensionality reduction techniques for Brain Computer Interfaces” by Tian Lan has been examined and approved by the following Examination Committee:

Peter A. Heeman
Associate Research Professor
Department of Biomedical Engineering, OHSU

Deniz Erdogmus
Assistant Professor
Northeastern University

Xubo Song
Associate Research Professor
Department of Biomedical Engineering, OHSU

Brian Roark
Associate Professor
Department of Biomedical Engineering, OHSU

Jennifer G.Dy
Associate Professor
Northeastern University

# Contents

# Acknowledgements

It is my pleasure to thank many people who made this thesis possible.

First and foremost I want to thank my advisor Prof. Deniz Erdogmus for the support of my Ph.D study and research, for his patience, enthusiasm, and inspiration. He provided lots of good ideas and advice throughout my study. He also helped me with paper writing skill.

I would also like to thank another advisor Prof. Jan van Santen for supporting me continuing my research in Brain Computer Interfaces. He also provided valuable advice along the research project.

Besides my advisors, I would like to express my sincere gratitude to Prof. Peter Heeman, who helped me with thesis writing and the graduation process, as well as the petition. My thesis would not be done without you.

I am indebted to many people who were involved in this process. Thank Prof. Xubo Song, Prof. Brian Roark, and Prof. Jennifer G.Dy for being my thesis committee members. Thank my colleague Catherine Huang for discussing the project and offering data and matlab code. Thank Pat Dickerson for helping with paperwork and other process. Thank Rebecca Lunsford and Ethan Selffridge for giving me great feedback on my practice talk.

Last but not least, I would like to thank my parents, my sisters, and my Fiancée Sumin, for encouraging and supporting me to finish this journey. I dedicate this thesis to you.

# Abstract

Brain Computer Interfaces (BCIs) refer to direct interactions between human brains and computers, which offer non-muscular communication and control channels. BCIs are particularly useful in some applications, such as sensorimotor control, neuromuscular disorders rehabilitation, task-related performance augmentation, and target recognition. Modern non-invasive BCIs use electroencephalography (EEG) to measure brain activities, and use various signal processing and machine learning techniques to interpret the results. From the machine learning point of view, a BCI system can be seen as a classification system, which contains five parts: 1) pre-processing; 2) feature extraction; 3) feature selection and dimensionality reduction; 4) classification; and 5) post-processing. However, BCI applications are highly data-dependent, and no universal solution exists for all applications to solve the robustness, realtime, and nonstationarity problems.

In this thesis, we propose that feature manipulations, including feature extraction, feature selection and dimensionality reduction, can solve or at least partly solve the robustness, realtime and nonstationarity problems. Our research focuses on two BCI applications: Augmented Cognition (AugCog) and single trial ERP detection.

Augmented cognition (AugCog) is an embryonic concept aiming at enhancing the task-related performance of human users through computer-mediated assistance based on assessments of cognitive states in real-time during the execution of certain tasks. In this application, we develop different linear and non-linear feature selection and dimensionality reduction methods using Mutual Information (MI) as the criterion. We also develop a statistical similarity based approach for feature extraction.

Single trial ERP detection aims at detecting Event Related Potential (ERP) after the stimuli onset, which can be used for target recognition. In the single trial ERP detection application, we compare different feature extraction, feature selection and dimensionality methods in the time, frequency and spatial domains.

Experimental results show that the proposed methods improve the performance of BCI systems compared with our baseline systems. For each method we present, we also discuss both advantages and disadvantages, and give general guidelines in selecting different techniques for different data structures.

# Chapter 1: Introduction

## 1.1 Brain Computer Interfaces

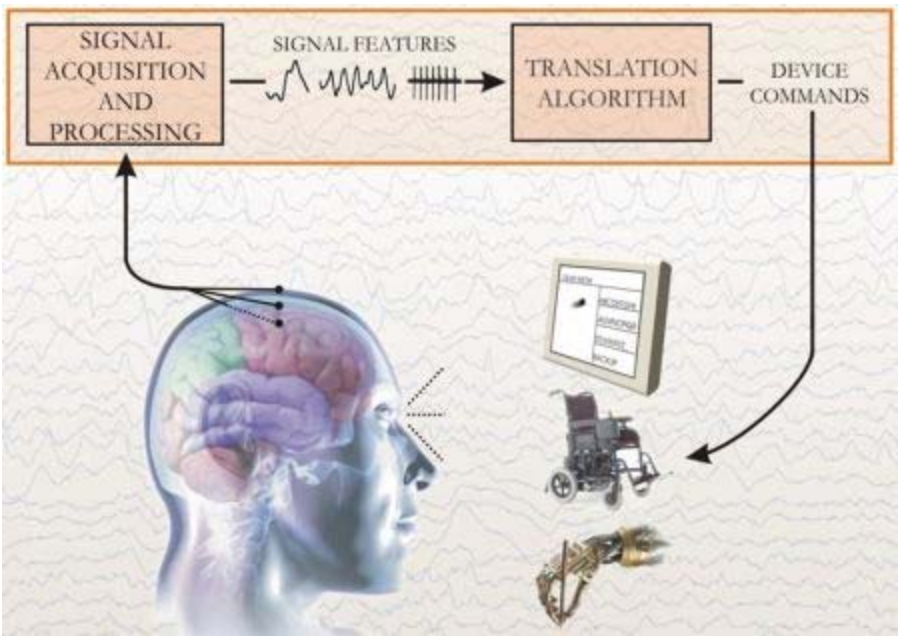



Figure 1-1 A typical BCI system (Picture courtesy of Gerwin Schalk, Wadsworth Center, NY)

Brain Computer Interfaces (BCIs) refer to a family of designs that facilitate direct interactions between human brains and computers. Unlike traditional Human Computer Interfaces (HCIs), BCIs offer new, non-muscular communication and control channels, which makes them particularly useful in some applications, such as assisting the disabled people, repairing human cognitive or sensory-motor functions, and augmenting task-related performance. A typical BCI system contains four parts (Figure 1-1): brain activity acquisition, signal preprocessing and feature extraction, mental state classification/estimation, and control output. Ever since Vidal (1973; 1977) showed that brain signals could be used to build a mental prosthesis,

non-invasive BCIs have attracted increasing interest. Many studies have been done in different areas. Wolpaw and McFarland (1994; 2004) showed that a non-invasive BCI that uses scalp-recorded electroencephalography (EEG) activity can provide humans with multidimensional movement control. Farwell and Donchin (1988) showed that the visual P300 Event Related Potential (ERP) can be used to select letters from a computer display.

### 1.2 EEG based BCI

Modern non-invasive BCI applications use electroencephalography (EEG) to measure brain activities. This is because that compared with other measurement modalities, such as magnetoencephalogram (MEG) and functional magnetic resonance imaging (fMRI), EEG is inexpensive, convenient, and has high temporal resolution. However, there are many challenges for EEG based BCIs due to the following factors in brain wave measurement:

(1) Noise resulting from motion artifacts;

(2) Contamination with muscular activities, including the eye movements and blinks;

(3) Influence of concurrent but irrelevant neural activities;

(4) Environmental noise from other electrical devices;

(5) Nonstationarity of the brainwave.

Many signal processing and machine learning techniques have been used to reduce the noise and artifacts of EEG signals successfully, such as adaptive filter and Independent Component Analysis (ICA) (Hillyard and Galambos, 1970; Whitton *et al.*, 1978; Li and Principe, 2006). Wavelet and other time-

frequency domain methods have also been proposed to deal with the nonstationary issue (Thakor *et al*. 1993; Schiff *et al*., 1994; Krystal et al., 1999).

### 1.3 BCI Applications

EEG based BCIs have a wide range of applications, such as sensorimotor control, neuromuscular disorders rehabilitation, augmented cognition, and target recognition using single trial ERP detection.

#### 1.3.1 Sensorimotor Control

Sensorimotor control is achieved by controlling the amplitudes of sensorimotor rhythms. The sensorimotor rhythms are 8-12Hz (mu) and 18-26Hz (beta) components in the EEG signals. The changes of mu and beta are associated with movements, sensation, and imagery movements. Previous studies have shown that subjects can learn to control mu and beta rhythm amplitudes in the absence of movement or sensation (Wolpaw *et al*., 1991; Wolpaw and McFarland, 1994; Wolpaw and McFarland, 2004). In experiments, scalp EEG was recorded over sensorimotor cortex and the amplitudes of sensorimotor rythms were decorded to map the left/right and up/down movements. In this way, one can control cursor movements on screen in one or two dimensions (Wolpaw *et al*., 1991). Complex control tasks, such as robot arm control, can also be achieved using a similar procedure.

#### 1.3.2 Neuromuscular Disorders Rehabilitation

Neuromuscular disorders are diseases that either directly or indirectly impair the functioning of the muscles. Traditionally, the communication and control functions can be restored by replacing neural signals with electrical simulation (Hoffer *et al*., 1996; Kilgore *et al*., 1997). Nowadays, another option using direct BCIs

is emerging. Direct BCIs allow patients to convert their thoughts into actions that do not involve voluntary muscle movements. The non-muscular channels that BCIs offer provide rehabilitation for either neuromuscular disorders or physical damages. For example, a paralyzed person can control the wheel chair by a BCI system: EEG signals are collected from the patient and then decoded to map the imagery movements to the wheel chair control signals.

### 1.3.3 Augmented Cognition

Augmented Cognition is a new born concept that aims at extending human performance via the help of computers. In a mental task, the performance is restricted by the limitation of a person's attention, memory, and comprehension abilities (Schmorrow and Kruse, 2004). An Augmented Cognition system can adaptively manage the information sent to users based on their mental states or workloads. A BCI system is used to evaluate and monitor the user's cognitive status in real time. An example of Augmented Cognition is in the battle field: a commander should assign a new task to solders whose workloads are low to maximize the group performance. Another example is a system that will reduce the information presentation rate when the user exhibits mental fatigue or distractive.

### 1.2.4 Target Recognition using Single Trial ERP Detection

An Event-Related Potential (ERP) is a series of peaks in the EEG signals when an unexpected, sparse stimulus presents. Thrope showed that the ERP can be used for target detection in a Rapid Serial Visual Presentation (RSVP) task, in which a series of images are presented to the subject at a fast speed (Thorpe *et al.*, 1996). Among all different types of ERPs, P300, which is a positive potential that happens around

300ms after the stimulus appears, has been extensively studied. Farwell and Donchin (1988) showed that the visual P300 ERP can be used to select letters from a computer display. They developed a speller that presented letters and digits in a 6×6 matrix, then highlighted rows and columns in a random sequence. The user "selected" the row and the column that contains the target letter consecutively by mind, thereby selected the intended letter.

## 1.4 Machine Learning Perspective of BCIs

From the machine learning perspective, a BCI requires a robust pattern classification system to assess cognitive states of human subjects. A typical classification system contains five components (Figure 1-2):

1) Pre-processing;
2) Feature extraction;
3) Feature Selection and dimensionality reduction;
4) Classification;
5) Post-processing.

An improvement in just one of these components can improve the performance of a BCI system, we focus our work on the component (2) feature extraction and (3) feature selection and dimensionality in order to acquire a set of compact and informative features. This not only enables us to find the intrinsic characters of brain activities, but also greatly reduces the computational load of the system, hence can be a solution to problems we described in Section 1.3. Our research focuses on different techniques of feature extraction, feature selection and dimensionality reduction, and their applications in both Augmented Cognition and single trial ERP detection.

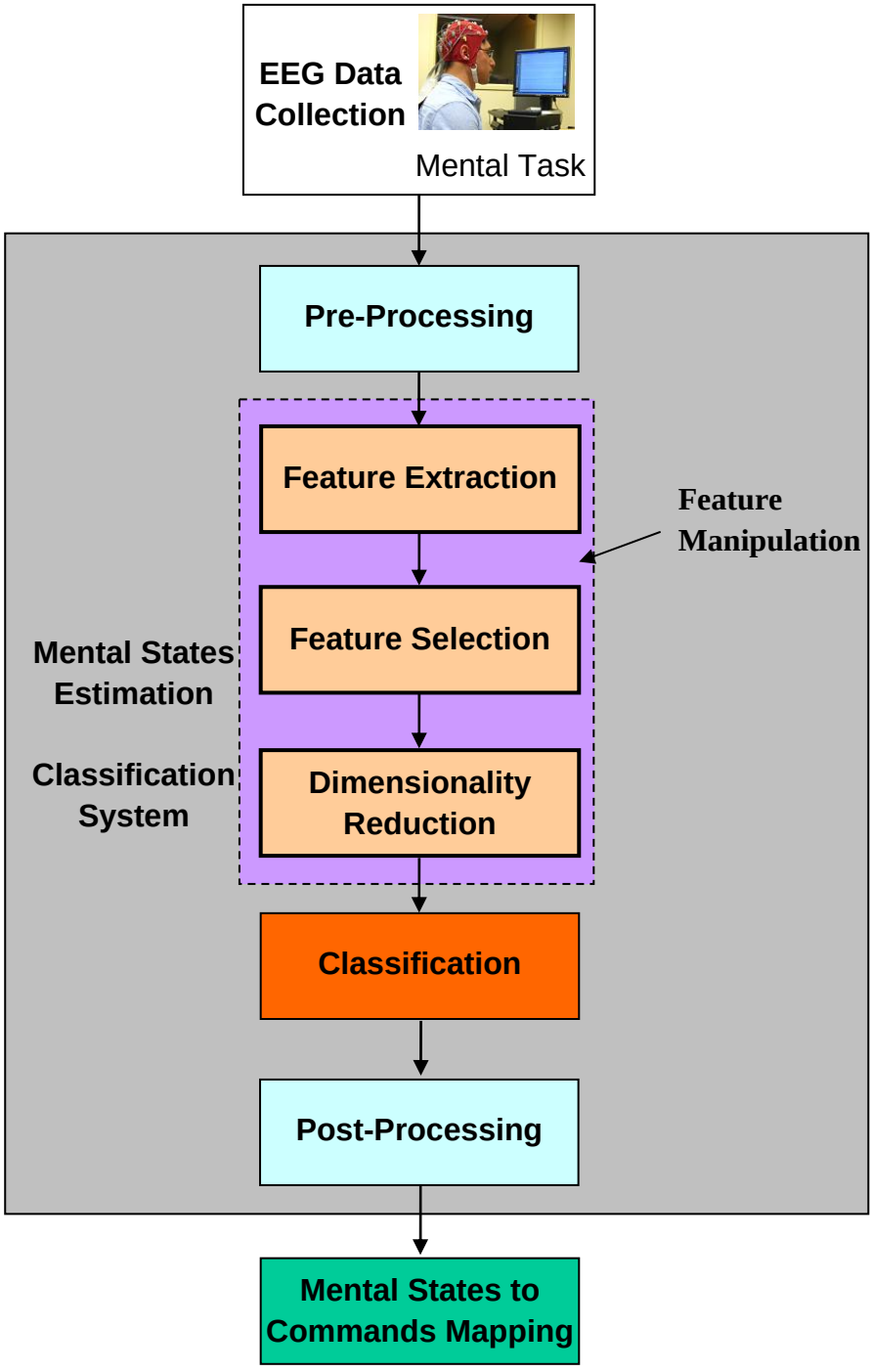



Figure 1-2 The block diagram of a typical Brain Computer Interface system

## 1.5 Problem Definition

Among EEG applications, we focus on two of them in this thesis: Augmented Cognition and single trial ERP detection. As mentioned in Section 1.2, there are many challenges in the EEG based BCIs. In practice, we need to solve the following problems:

1) The robustness of the system. The system should be able to deal with noise from both environment and subject muscular activities, as well as to be used in mobile environments.
2) The nonstationarity of the system. EEG can be treated as stationary signal within a short time. However, due to the nonstationarity of brain activities, the system needs to be re-trained after a certain period of time. To be specific, the system should have the abilities of Session-to-Session transfer, which means a system trained in one session can be used in another session, and Subject-to-Subject transfer, which means a system trained on one subject can be used on another subject.
3) The system should run in realtime. Realtime requirement is critical to many real-world closed-loop BCI systems.

In the next section, we will analyze the BCI system from a machine learning perspective, and propose solutions to the above problems.

## 1.6 Thesis Statements

In summary, we have the following two thesis statements:

First, feature manipulations, including feature extraction, feature selection and dimensionality reduction, can solve or at least partly solve the three problems we introduced in Section 1.5: 1) robustness; 2) nonstationarity; and 3) realtime. A set of compact and informative features eliminate most of the

irrelevant interruptions and noise, and hence makes the system more robust. They also greatly reduce the computational load of the system, and allow us to do online training and classifying. These features also capture the intrinsic characters of brain activities related to a certain mental task. Therefore training a classification system using these features will at least partly solve the Session-to-Session transfer and Subject-to-Subject transfer problems. Even after long time we need to re-train the system, since the system can work in realtime, training time can be negligible.

Second, no universal feature manipulation method exists to deal with different applications, because BCIs are highly data-dependent. Even for the same type of applications, when subjects execute different mental tasks which may be related to different brain activities, different feature manipulation techniques are needed for the best performance.

## 1.7 Thesis Contributions

There are two major contributions of the thesis. First, we propose different feature extraction, feature selection and dimensionality reduction methods and apply them in two of BCI applications to solve the robustness, nonstationarilty and realtime problems. Second, we comprehensively compare the results of different feature manipulation methods in different BCI applications, and give general guidelines in selecting techniques.

With respect to the thesis contribution on the first BCI application, Augmented Cognition, there are three related parts. First, we propose a mutual information (MI) based framework, linear ICA-MI for feature selection and dimensionality reduction. We use the piece-wise linear to approximate non-linear solution and generalize the linear ICA-MI framework into non-linear domain. Second, we propose another

mutual information based non-linear feature selection framework, GMM-MI, estimating the mutual information using Gaussian Mixed Model (GMM), and compare this method with ICA-MI method. Third, we propose a statistic similarity based feature extraction method, and compare it with the traditional Power Spectrum Density (PSD) integration method in Augmented Cognition.

With respect to the thesis contribution on the second BCI application, single trial ERP detection, there are two related parts. First, we explore and compare different feature extraction, feature selection method in the time, frequency and spatial domains. Second, we propose a channel-wise dimensionality reduction scheme for the EEG channel selection and compare different subspace projection methods for the channel-wise dimensionality reduction in single trial ERP detection.

## 1.8 Thesis Structure

In Chapter 2, we give a literature review of the signal processing and machine learning techniques used in current BCIs. Then the thesis is divided into two parts based on two BCI applications: Augmented Cognition and single trial ERP detection.

Part one includes Chapter 3-6. In Chapter 3, we introduce the concept of Augmented Cognition and present a baseline BCI system for it. In Chapter 4, we present a mutual information based framework, linear ICA-MI, for feature selection and dimensionality reduction. In Chapter 5, we present two non-linear mutual-information-based feature selection methods: a piece-wise linear method and a GMM (Gaussian Mixed Model) method. In Chapter 6, we present a similarity-based feature extraction method.

Part two includes Chapter 7-10. In Chapter 7, we introduce the concept of single trial ERP detection and present a baseline system for it. In Chapter 8, we present a temporal windowing scheme to select the

features in the time domain. In Chapter 9, we present the feature selection methods in the frequency domain and the spatial domain. We also introduce the concept of channel-wise dimensionality reduction using subspace projection to reduce the computational load of EEG channel selection. In Chapter 10, we compare different subspace projection methods for the channel-wise dimensionality reduction.

In Chapter 11, we summarize the work we have done in this thesis, discuss the limitation of our current techniques and propose future research work.

# Chapter 2: Background and Related Work

## 2.1 Introduction

Ever since the first experimental demonstration in 1999 that ensembles of cortical neurons could directly control a robotic manipulator (Chapin *et al*., 1999), BCI has attracted increasing interest. BCI applications have been developed for sensorimotor control, neuromuscular rehabilitation, ERP detection, and Augmented Cognition. EEG based non-invasive BCI has become more and more popular, with more EEG electrodes used to collect data, from 8 to 16, and now up to 256 electrodes. As mentioned in Section 1.4, a typical BCI system can be treated as a classification system from a machine learning perspective. Different methods for each component of BCI systems have been developed in the past decades. In this chapter, we will give a brief review of signal processing and machine learning techniques used in each component.

## 2.2 Pre-processing

Typically, the non-invasive EEG recording has low Signal-to-Noise Ratio (SNR) due to the environment noise and artifacts caused by muscle and eye movements. The goal of pre-processing is to increase SNR. A variety of options for improving BCI SNR have been studied, including filtering in the frequency and spatial domains, normalization, and artifact removal.

### 2.2.1 Filtering

Filtering can be applied in the frequency domain. By selecting different pass bands, frequency filtering removes unwanted noises, such as wandering direct current and high frequency noise (1-45Hz). Frequency

filtering also has the ability to select relevant frequency components, such as sensorimotor rhythm 8-12Hz (mu).

Filtering can also be applied in the spatial domain. The goal of spatial domain filtering is to create a subset of EEG channels which are related to certain brain activity, as well as enhance the separability of the data. The choice of spatial filter can affect the SNR greatly (McFarland *et al*., 1997). The simplest spatial filter is the bipolar derivation, which derives the first spatial derivative and enhances differences in the gradient in one direction. A Laplacian derivation is the second derivation of the instantaneous spatial voltage distribution, which emphasizes activities in radial sources immediately below the recording location (Nunez *et al*., 1994; Nunez *et al*., 1997). Other spatial filters are also available. Principal components analysis (PCA), independent components analysis (ICA) and common spatial patterns analysis are alternative methods for deriving weights for a linear combination of EEG channels (Muller-Gerking et al., 1999; Jung *et al*., 2000).

In this thesis, we use the time domain filtering for both applications for its simplicity and performance.

### 2.2.2 Normalization

Normalization is used to scale the data in each channel to zero mean and unit standard deviation. During EEG data collection, the resistance of each EEG electrode changes over the time, which causes changes of voltage readings. As a result, variances of some EEG channels can be significantly different to those of others. Even for the same EEG channel, the signal drifts over time. Normalization process not only improves the classification performance, but also solves the nonstationary problem within one session. We used the normalization process in our single trail ERP detection system.

### 2.2.3 Artifacts Removal

Muscle activation and eye movement can contribute to the electrical activity recorded from the scalp (Anderer *et al*., 1999; Croft and Barry, 2000). Different methods are available to deal with artifacts removal. Methods based on Principal Components Analysis (PCA) only rely on second order statistics of the data, and can be implemented in realtime. However, PCA methods require the orthogonal topographies in the spatial domain, which is not always the case. Similarly, Independent Components Analysis (ICA) can cleanly separate artifacts from brain activities using higher order statistics of the data; however, this method assumes EEG is derived from a limited set of spatially stationary sources (Jung *et al*., 2000), which is not true in most of cases. Furthermore, methods based on ICA are usually computationally intensive, hence are not suitable for realtime systems. Perhaps the most widely used method is adaptive filtering, which removes the artifacts by subtracting the estimated noise (Widrow and Hoff, 1960). Usually, eye blinks are recorded by vertical and horizontal Electro-Oculogram (EOG) channels independent of EEG channels. The eye blink artifacts of each EEG channel are estimated from VEOG and HEOG. The drawback of this method is that the EOG is not only sensitive to eye artifacts, but also contains brain activity from the frontal cortex. EOG subtraction can result in a considerable distortion of relevant brain signals (Berg and Scherg, 1994). This drawback can be reduced by filtering the EOG signals with a low pass filter at 8Hz (Jervis *et al*., 1988). In our thesis, we use the adaptive filtering method for artifacts removal in the AugCog system.

## 2.3 Feature Extraction

The problem of feature extraction is to find intrinsic characteristics from EEG signals that are related to certain mental activity. While pre-processing can remove the muscle and eye movement artifacts, feature

extraction can partly remove the CNS (central nervous system) artifacts, which are related to irrelevant neuron activities. Feature extraction methods can greatly affect the SNR, which determines the performance of a BCI.

### 2.3.1 Time Domain Feature Extraction

Feature extraction can be achieved in the time domain. Event Related Potentials such as P300 are widely used time domain features for single trial ERP detection (Farwell and Donchin, 1988; Donchin and Coles, 1988; Donchin *et al*., 2000). Autoregressive (AR) based feature extraction is a commonly used method, where the AR coefficients are used as features of the EEG signal. The AR modeling approach has attracted a lot of interest in BCI (Obermaier *et al*., 2001) since it does not involve meticulous tuning of subject-specific frequency bands. In our single trail ERP detection system, we use time domain features because they have better temporal resolution.

### 2.3.2 Frequency Domain Feature Extraction

Feature extraction can also be achieved in the frequency domain. In sensorimotor control, the amplitudes of mu (8-12Hz) and beta (18-25Hz) are used to map thoughts to one or two-dimensional movement (Wolpaw *et al*., 1991; Wolpaw and McFarland, 1994; Wolpaw and McFarland, 2004). In other BCI applications, a particular selection of the frequency bands is used based on well-established interpretations of EEG signals in prior cognitive and clinical contexts (Gevins *et al*., 1997).

### 2.3.3 Time-Frequency Domain Feature Extraction

A BCI could conceivably use both the time domain and the frequency domain features to improve performance (Schalk *et al*., 2000). A local or short-time Fourier transform (STFT) provides a degree of temporal resolution by highlighting changes in spectral response with respect to time. However, the Fourier integral might potentially obscure transient of location-specification features within the signal due to the average effect. This limitation can be partly overcome by introducing a fixed length sliding window, or using wavelet transforms (Vidal, 1977; Middendorf *et al*., 2008; Sperlich and Hillyard, 2004). Wavelet transforms produce a time frequency decomposition of a signal over a range of characteristic frequencies that separates individual signal components more effectively than do STFTs. However, in a lot of applications, where the energy level of certain frequency bands are preferred, the power spectral density (PSD) of the EEG signals, estimated using the Welch method (Welch, 1967), is more suitable. We use PSD as features in our AugCog system because the energy level in different frequency range is related to the subjects' workload.

## 2.4 Feature Selection and Dimensionality Reduction

Feature selection and dimensionality reduction are essential steps in a classification system. Generally speaking, reducing the number of features can be achieved by feature selection, in which a subset of features is picked to maximize the classification accuracy, and/or dimensionality reduction, in which new, low dimensional features are derived from the input features. It is desirable to keep the dimensionality of features as low as possible, not only to reduce the computational load, but also to make the system robust (Torkkola, 2003; Murillo and Rodriguez, 2007). For example, Chow and Huang (2005) showed that training a classifier using a small data set with high dimensionality greatly degrades classification

performance. Reducing the number of dimension of raw features before sending them to a classifier is a widely used technique in classification systems. In the past years, many linear and non-linear feature selection and dimensionality reduction methods have been developed to deal with different conditions.

### 2.4.1 Feature Selection

There are two types of feature selection methods: wrapper approach and filter approach. In wrapper approach, the feature selection process is coupled with a specific classifier. The feature selection criterion is to minimize classification error, or Bayes cost. In filter approach, the goal is to optimize a criterion that is related to Bayes risk, but independent of classifiers. For example, Mutual Information (MI) between input features and class labels is a widely used criterion for filter approach because it is related to the upper and lower bounds of Bayes error (Fano, 1961; Hellman and Raviv, 1970). On the other hand, the wrapper approach is optimal to a selected classifier, but is not flexible. Moreover, the selection process involves combination complexity of training and testing, hence is very computation intensive, and is not feasible for online systems. When using a cluster of classifiers, the filter approach is generally preferred for the low computation cost. In this thesis, we use the filter approach in the AugCog system for its flexibility, and use the wrapper approach in the single trial ERP detection system for its performance.

### 2.4.2 Dimensionality Reduction

Usually, dimensionality reduction is achieved by subspace projection. There are many existing linear subspace projection methods and their non-linear version. Principle component analysis (PCA) is a widely used dimensionality reduction technique (Oja, 1983; Devijver and Kittler, 1982). However, the projections

it finds are to maximize variances, which are not necessarily related to classification performance. Linear discriminant analysis (LDA) attempts to eliminate this shortcoming of PCA by finding linear projections that maximize class separability under the Gaussian distribution assumption (Fukunaga, 1990). The LDA projections are optimized based on the means and the covariance matrices of classes, which are not descriptive of an arbitrary probability density function (pdf). Independent component analysis (ICA) has also been used as a tool to find linear transformations that maximize the statistical independence of random variables (Everson and Roberts, 2003; Hyvärinen, A., *et al*., 1998). However, it has similar drawbacks as PCA (Torkkola, 2003).

Similar to feature selection, MI between projected features and class labels can be used as a criterion to optimize subspace projections. Hild *et al*. (2006) developed a feature projection method called MRMI which used Renyi's entropy to approximate Shannon's entropy. This method overcomes the drawback of PCA, ICA and LDA. However, it is computational intensive and not suitable for a realtime system. In this thesis, we use and compare different dimensionality methods in different applications.

## 2.5 Classification

In this section, we review a few most popular and widely used classifiers in BCIs.

### 2.5.1 Linear Discriminant Analysis (LDA) Classifier

LDA uses a hyperplane to separate different classes (Duda, 2000; Fukunaga, 1990). The hyperplane is obtained by finding the projection that maximizes the distance between the class means and minimizes the classes variance. Due to the low computational cost, LDA classifiers have been widely used in realtime

BCI systems. Successfully useed LDA in sensorimotor control or single trial ERP detection has been reported (Pfurtscheller, 1999; Bostanov, 2004). The main shortcoming of LDA is that it can not deal with non-linear data. We use the LDA classifier in our single trial ERP detection system.

### 2.5.2 Support Vector Machine (SVM) Classifier

An SVM classifier uses a discriminant hyperplane to separate the classes. Unlike LDA, the hyperplane is selected to maximize the margins, which is the distance from the nearest support vectors (Burges, 1987). It is possible to extend linear SVM into non-linear domains by playing the 'kernel trick', which maps a non-linear kernel function into another space. Usually, a Gausian kernel is used. SVM classifiers have better generalization performance, and are resistant to the curse-of-dimensionality. However, the parameters in an SVM need to be trained using cross validation to maximize the generalization performance, which is a long process, or be chosen based on experience. SVM classifiers are widely used in BCI applications. In our thesis, we use an SVM classifier in our single trial ERP detection system.

### 2.5.3 MultiLayer Perceptron (MLP) Neural Networks

An MLP is composed of multiple layers of neurons (Bishop, 1996). An MLP is a universal approximator. When using enough neurons and layers, it can approximate any continuous function. This make MLP classifiers suitable for almost any classification problem, hence are widely used in BCIs (Palaniappan, 2005; Anderson and Sijercic, 1996). However, MLP classifiers are sensitive to overfitting and outliers, especially with noisy and non-stationary data. Therefore, MLP classifiers may not be good choices for a robust system. We do not use the MLP classifiers in our BCI applications.

### 2.5.4 Hidden Markov Model (HMM)

HMMs are popular dynamic classifiers in speech recognition (Rabiner, 1989). HMMs are also suitable for classifying time series. HMMs have been applied to classifying of temporal sequences of BCI features (Obermaier, 2001; Cincotti *et al*., 2003). However, the performances of HMM classifiers are inconsistent. For example, the results in Cincotti *et al.* (2003) are not comparable with other classifiers, which are probably due to the data structures. HMM classifiers are rarely used in BCI applications.

### 2.5.5 K-Nearest Neighbor (KNN) Classifier

KNNs are non-parametric classifiers that make no assumption about the form of the probability densities underlying a particular set of data. The classifier assigns an unseen point to the dominant class among its k nearest neighbors within the training set. It can be shown that if *K* is large, this classifier approaches the best possible classification performance given by the true Bayes classifier (Duda, R.O., *et al*., 2000). KNN classifiers are rarely used in the BCI community because they are sensitive to the curse-of-dimensionality. However, when working with low dimensional features, KNN classifiers are very efficient. We use the KNN classifier in our AugCog system.

### 2.5.6 Combinations of classifiers

There are a lot of other classifiers that can be used for BCIs. More and more BCI applications are starting to use more than one classifier. Using a combination of classifiers can yield better performance. For example, three different classifiers can be used separately on BCI data, and the final result can be fused using majority vote from three classification results. Another way of combining them is to use a cascade: in

level 0, multiple classifiers are applied on the whole data, or a subset of the data. The outputs of the level 0 classifiers are used as inputs for the level 1 classifier(s). In our AugCog system, we do a majority vote on a committee of three classifiers. And in our single trial ERP detection system, we compare a cascade of classifier with a single SVM classifier.

## 2.6 Post-processing

Post-processing uses context information to eliminate outliers, hence to improve the performance of the classifiers. For example, in robot arm control, the control signal must be continuous. This constraint is used as feedback to tune the classifier. Post-processing is highly application dependent. No universal rule exists to deal with all situations. In our AugCog system, we use a median filter for post-processing.

## 2.7 Summary

In this chapter, we reviewed the signal processing and machine learning techniques that are currently used in BCIs. The design of a BCI system requires us to choose a certain technique for each component based on characteristics of the BCI task. In the next chapter, we will introduce one BCI application: Augmented Cognition (AugCog). We will introduce the concept and development of AugCog, and a baseline system that uses techniques we mentioned in this chapter. In the chapters following that, we will focus on different feature extraction, feature selection and dimensionality reduction methods. We follow that to introduce the other BCI application, single trial ERP detection, in Chapter 8.

# Chapter 3: Augmented Cognition

Starting in this chapter, we present our work on one the of the BCI applications: Augmented Cognition. This chapter focuses on a baseline system for Augmented Cognition. In Chapter 4 to Chapter 6, we focus on feature extraction, feature selection and dimensionality reduction techniques.

Augmented Cognition (AugCog) is an embryonic concept aimed at investigating the feasibility of using psychophysiological measures of cognitive activity to guide the behavior of human–computer interfaces (Kruse and Schmorrow, 2005). The goal of AugCog is to enhance the task-related performance of a human user through computer-mediated assistance based on assessments of cognitive states in real-time during the execution of certain tasks. The AugCog concept has the potential to revolutionize the way that human operators (e.g., drivers, machinery operators, surgeons, soldiers) interact with their environments, as computer systems manipulate the human information processing requirements based on the real-time assessment of workload and performance.

An essential component of an AugCog system is a brain-computer interface (BCI), as its operation is controlled by the estimates of mental states from sensed brain activity signals. BCI refers to a family of engineered systems that modulate the interactions between human brains and computers. In most current experimental AugCog systems, the instantaneous estimates of mental state and workload are used to control the rate and the modality of the information presented to the operator. In this way, the cognitive resources of the operator can be allocated to maximize cognitive performance (Pavel *et al*., 2003). However, unlike current BCI approaches, an important characteristic of AugCog systems is to facilitate cognitive state estimation based on ambulatory electroencephalogram (EEG), which allows prolonged EEG recording in

the home setting. On the contrary, user mobility is strictly constrained in traditional BCI contexts. Therefore, motion artifacts and potential nonstationarity present an additional challenge in terms of preprocessing.

What's more, inferring mental states from one subject's EEG signals to another subject's EEG signals, the so-called subject-to-subject transfer, or inferring mental states for the same subject from one session EEG signals to another session EEG signals, as called session-to-session transfer, are another two challenges to AugCog as well as other BCI applications.

The use of EEG as the basis of assessment in BCI and AugCog systems is predicated on characteristics such as good temporal resolution, low invasiveness (relative discomfort), low cost, and portability. However, the following factors make it particularly difficult to deal with ambulatory EEG signals: (1) noise resulting from motion artifacts; (2) contamination with muscular activities, including the usual eye movements and blinks; (3) influence of concurrent but irrelevant neural activities; (4) environmental noise; (5) nonstationarity. Under these circumstances, both robustness and precision of the designed AugCog system are particularly critical. Furthermore, the system needs to be portable and be able to work in real-time in some of mental tasks. In this chapter, we will present a baseline AugCog system that focuses on solving these problems. The base system contains data collection, signal processing, and cognitive state classification. Part of this chapter was previously published in (Lan *et al*., 2005a).

## 3.1 EEG Data Collection

In AugCog system, EEG signals are collected from human subjects' scalps using non-invasive method when subjects execute certain mental tasks. Different subjects repeated two mental tasks, Larson and n-back (Halgren *et al*., 2002; Gevins *et al*., 1997; Gevins and Smith, 2000), in different days.

### 3.1.1 Mental Tasks

In the Larson task, the subjects are required to maintain a mental count according to the presented configuration of images on the monitor. The combination of mental activities during this task includes *attention*, *encoding*, *rehearsal*, *retrieval*, and *match*. The complexity of this task was manipulated by varying the inter-stimulus interval (low and high).

In the n-back task, subjects are required to match the letter in either spatial location or verbal identity in the previous trials. The easy task only requires comparing the current stimuli with the first one, involving the combination of mental activities including attention, and match. The difficult task requires comparing the current stimuli with stimuli presented two trials previously, and involves a complex combination of mental activities that includes attention, encoding, rehearsal, retrieval, and match.

### 3.2.2 Hardware Platform

EEG data is collected using the BioSemi Active Two system using a 32 channel EEG cap and eye electrodes (Figure 3-1). This system integrates an amplifier with an Ag-AgCl electrode, which affords extremely low noise measurements without any skin preparation. This unit senses a subject's ECG signals and outputs inter-beat interval data in conjunction with a derived measure of a subject's cognitive arousal. Information from the sensors described above is processed on a body worn laptop. The mobile data

computer is a Dell laptop. The sensors are connected via a combination of USB, serial port and Bluetooth. Sensor data is collected and processed on the mobile data processing computer during the experiment. A base station computer controls the experiment and communicates with the mobile data processing computer via an 802.11 wireless network.

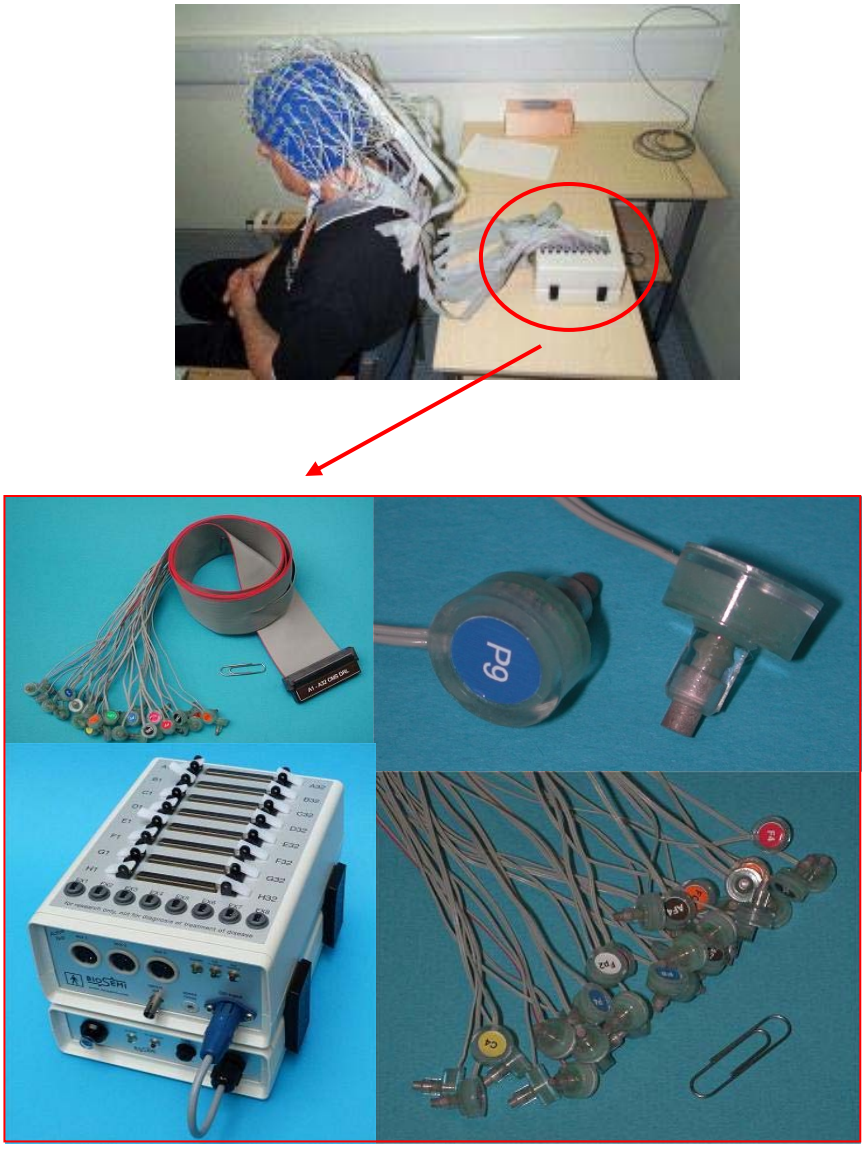



Figure 3-1 Bio-Semi system hardware

### 3.2 Signal Processing Module

In a typical EEG based BCI system as showed in Figure 1-2, the signal processing module usually refers to the part that converts the raw EEG signals to usable feature vectors which are used for classification. In our baseline system, the signal processing module only contains pre-processing and feature extraction. We will not discuss feature manipulation in this chapter, which will be covered in Chapter 4 and 5.

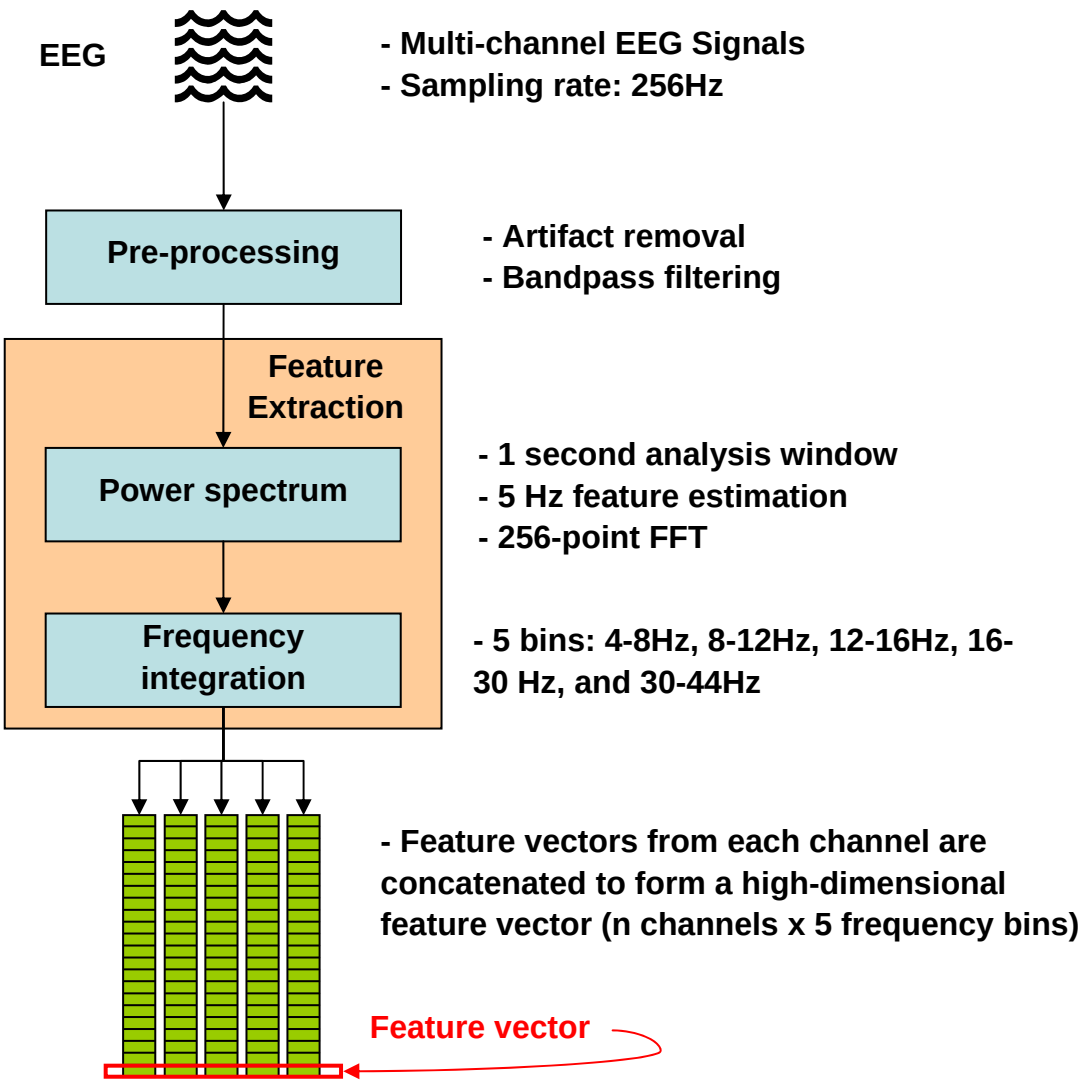



Figure 3-2 Signal Processing Components in the baseline system

### 3.2.1 Pre-processing

The cognitive state classification efforts reported here rely primarily on EEG data. As mentioned earlier, the monitoring equipment consistes of a BioSemi Active Two EEG system with 32 electrodes (also called EEG channels). Vertical and horizontal eye movements and blinks are recorded with electrodes below and lateral to the left eye. All channels reference the right mastoid. EEG is recorded at 256Hz sampling frequency from all channels while the subject is performing tasks. Although all channels are recorded, only 7 channels (CZ, P3, P4, PZ, O2, P04, F7) are used for mental state estimation due to the limitation of wireless bandwidth. These channels were selected based on a saliency analysis on EEG collected from various subjects performing cognitive test battery tasks (Russell and Gustafson, 2001). EEG signals are pre-processed to remove eye blinks using an adaptive linear filter based on the Widrow-Hoff training rule (Widrow and Hoff, 1960). Information from the VEOGLB ocular reference channel was used as the noise reference source for the adaptive ocular filter. DC drifts were removed using high pass filters (0.5Hz cut-off). A band pass filter (between 2Hz and 50Hz) was also employed, as this interval is generally associated with cognitive activity.

### 3.2.2 Feature Extraction

The power spectral density (PSD) of the EEG signals, estimated using the Welch method (Welch, 1967) with 50%-overlapping 1-second windows, is integrated over 5 frequency bands: 4-8Hz (theta), 8-12Hz (alpha), 12-16Hz (low beta), 16-30Hz (high beta), 30-44Hz (gamma). These bands, sampled every 0.1 seconds, are used as the basic features for cognitive classification. The particular selection of the frequency

bands is based on well-established interpretations of EEG signals in prior cognitive and clinical (Gevins *et al*., 1997) contexts. The overall schematic diagram of the signal processing system is shown in Figure 3-2.

## 3.3 Cognitive State Classification

Estimates of spectral power form the input features to a composite classifier. The classifier uses parametric and non parametric techniques to asses the likely cognitive state (high and low workload) on the basis of spectral features, i.e. to estimate *p*(cognitive state | spectral features). The classification process relies on probability density estimates derived from a set of spectral samples. These spectral observations are gathered in conjunction with tasks representative of the eventual performance environment. It is assumed that these sample patterns are representative of the spectral patterns one would expect in the performance environment. The classifier uses three distinct classification approaches: K nearest neighbor (KNN), Parzen Windows, and Gaussian Mixture Models as shown in Figure 3-3. We describe each of these components in the following.

### 3.3.1 Gaussian Mixture Models (GMM) Classifier

Gaussian Mixture models provide a way to model the probability density functions of features associated with specific classes using a superposition of Gaussian kernels. The unknown probability density associated with a specific class is approximated by a weighted linear combination of Gaussian density components. Given, an appropriate number of components and appropriately chosen component parameters (mean and covariance matrix associated with each component), a Gaussian mixture model can model any probability density to an arbitrary degree of precision.

The parameters associated with Gaussians component are determined iteratively using the Expectation Maximization algorithm (Dempster A.P., *et al*., 1997). Once the Gaussian parameters have been initialized, the system iterates through a two step procedure for each sample associated with each class. In the first step (i.e. expectation step), the system computes the probability that a particular training sample belongs to a particular class based on the current model parameters (posteriori probability). In the maximization step, the model parameters are adjusted in the direction of increased class membership likelihood.

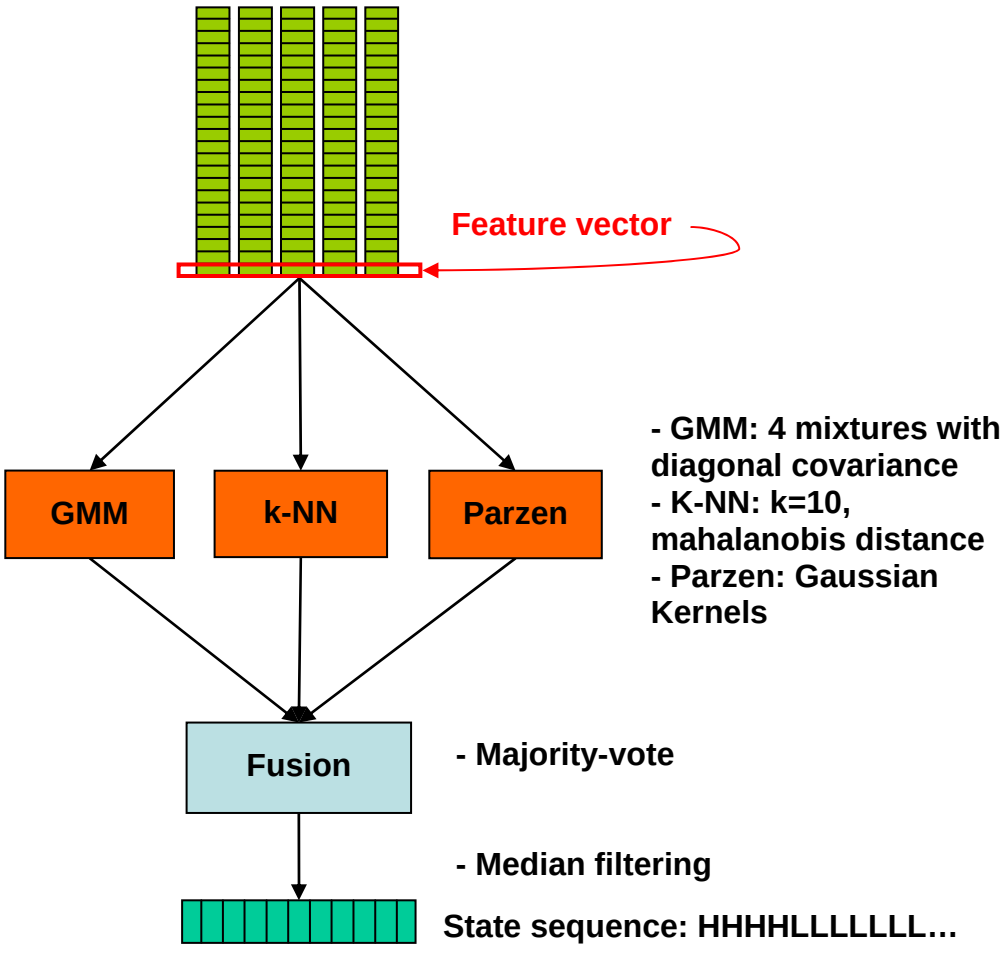



Figure 3-3 Classification system

### 3.3.2 K-Nearest Neighbor (KNN) Classifier

The K-nearest neighbor approach is a non-parametric technique that makes no assumption about the form of the probability densities underlying a particular set of data. Given a particular sample, the *K* nearest training samples (as assessed by Euclidian or Mahalanobis distance metrics) are determined and the test sample is assigned to the class which leads the most neighbors to this set. It can be shown that if *K* is large, this classifier will approach the best possible classification performance given by the true Bayes classifier (Duda, R.O., *et al*., 2000).

### 3.3.3 Parzen Window Classifier

Parzen windowing (Parzen, 1967) is a nonparametric density estimation technique. It is employed to estimate the class distributions and to form a nonparametric approximation to the Bayes classifier. In this context, it serves as a bridge between the KNN where each sample contributes discretely to the decision (depending on whether they are in the neighborhood or not) and the GMM classifier where each sample indirectly contributes to the Gaussian modes. In our implementation, we used Gaussian window functions, thus the Parzen classifiers is essentially a KNN classifier with decreasing influence by distance. At the same time it is a Gaussian mixture model itself, where a Gaussian is placed on each sample.

### 3.3.4 Composite Classifier

GMM, KNN and Parzen window classifiers were chosen over multi-layer neural networks because they require minimal training time. KNN and Parzen Windows require no training, whereas the EM algorithm is used to generate GMMs, converges relatively quickly. KNN and Parzen Window approaches require all

training patterns to be held in memory. Every new feature vector has to be compared to each of these patterns. However, despite the computational cost of these comparisons at run time, the system was able to output classification decisions well within real-time constraints.

The composite classifier treats the output from each classifier as a vote for the likely cognitive state. The majority vote of the three component classifiers forms the output of the composite classifier. When there is no majority agreement, the Parzen window decision is selected. A classification decision is generated at a rate of 10Hz. Outputs from the composite classifier are passed through a median filter before an assessment of cognitive state is output by the classification system, in order to make the cognitive state assessment process more robust to undesirable fluctuations in the underlying EEG signal. Modal filtering is done over a sliding 2 second window with the assumption that cognitive state remains stable over 20 decisions.

## 3.4 Experimental Results

The system described above was empirically assessed. The experiment compared classification accuracy across two workload levels in two mental tasks: Larson and n-back (Halgren *et al*., 2002; Gevins *et al*., 1997; Gevins and Smith, 2000) for three subjects. The EEG signals were collected, pre-processed and the features were extracted as described in Section 3.1 and 3.2. After that, the extracted features were sent to a committee of three classifiers for cognitive state estimation.

The classification results are showed via the confusion matrix (The $ij^{th}$ entry of confusion matrix **P** shows $P$(decide class $i$ | true class is $j$)) and total classification accuracy, as listed in Table 3-1. From the table we can see, the overall classification accuracy is about 70% for all subjects in two mental tasks.

Consider the classification is only two-class problem, in which a sample random guess can give us 50% overall accuracy, the results from our baseline system are not satisfactory.

Table 3-1 Classification results for two mental tasks of three subjects.

| | Larson | | n-Back | |
|---|---|---|---|---|
| | Confusion Matrix | Accuracy | Confusion Matrix | Accuracy |
| Subject 1 | 0.9835 0.0165<br>0.3427 0.6573 | 0. 7387 | 0.6641 0.3359<br>0.1059 0.8941 | 0. 7779 |
| Subject 2 | 0.6104 0.3896<br>0.3457 0.6543 | 0. 6436 | 0.1881 0.8119<br>0.0921 0.9079 | 0. 7286 |
| Subject 3 | 0.8222 0.1778<br>0.6354 0.3646 | 0. 4761 | 0.4853 0.5147<br>0.4362 0.5638 | 0. 5537 |

## 3.5 Discussion

In this chapter, we briefly introduced one of the BCI applications: Augmented Cognition. A baseline system is proposed to estimate cognitive mental states from human EEG signals when subjects were executing different mental tasks. Experimental results showed that the baseline system can classify the different level of workloads even though the EEG signals are noisy and non-stationary. However, the classification accuracy is not satisfactory. In the next three chapters, we will focus our study on different feature manipulation techniques, including feature extraction, feature selection and dimensionality, to extract more informative features to improve the classification accuracy. We also propose that by eliminating redundant features, the remaining features are more robust and hence partly solve the subject-to-subject transfer, session-to-session transfer, and non-stationary problems.

# Chapter 4: Linear Methods for Feature Selection and Dimensionality Reduction

In Chapter 3, we introduced a baseline AugCog system, which contains signal pre-processing, feature extraction, and classifiers. The baseline system estimates the subject's mental states based on the EEG inputs. However, in real world applications, we need better classification accuracy. Better classification accuracy can be acquired by improving any part of the classification system in Figure 1-2. We focus our work on feature selection and dimensionality reduction, since using a set of compact and informative features not only can meet the real time requirement, but also can partly solve the non-stationary challenge mentioned in Chapter 1. In this chapter, we introduce linear methods for feature selection and dimensionality reduction. All the work in this chapter was previously published in Lan *et al* (2005b; 2005c; 2007).

## 4.1 Background

Feature selection and dimensionality reduction are important steps in pattern recognition tasks and many other applications. In practice, the relevant information about the data structure can often be represented by a lower dimensional manifold embedded in the original Euclidian data space. Specifically, in pattern recognition, even when a high dimensional feature vector is available, usually the classification task can be achieved equally well by a feature vector of reduced dimensionality. Furthermore, reducing the number of features will also help the classifier learn a more robust solution and achieve a better generalization performance, since irrelevant feature components are eliminated by the optimal subspace projection.

### 4.1.1 Linear Projection vs. Non-linear Projection

Feature selection and dimensionality reduction by subspace projection is typically achieved by feature transformation methods. This transformation generates either a new feature space, or a sub-set of the original feature space, which can be treated as a special case of the former situation. The transformation can be linear or non-linear. Linear transformations have been widely used due to their simplicity. While nonlinear transformations are attracting increasing attention due to their ability to capture the nonlinear relationships within the data, the complexity of finding robust regularized nonlinear transformations makes them a second choice for most applications. In this chapter, we focus on linear transformations, leaving nonlinear transformations for the next chapter.

There are many existing linear transformation methods. Principle component analysis (PCA) is a widely used dimensionality reduction technique (Oja, 1983; Devijver and Kittler, 1982). However, since the projections it finds are not necessarily related to the class labels, it is not particularly useful in pattern recognition. Linear discriminant analysis (LDA) attempts to eliminate this shortcoming of PCA by finding linear projections that maximize class separability under the Gaussian distribution assumption (Fukunaga, 1990). The LDA projections are optimized based on the means and the covariance matrices of classes, which are not descriptive of an arbitrary probability density function (pdf). Independent component analysis (ICA) has also been used as a tool to find linear transformations that maximize the statistical independence of random variables (Everson and Roberts, 2003; Hyvärinen, A., *et al*., 1998). However, similar to PCA, the projection that ICA finds is also not necessarily related to the class labels, which may not be able to enhance class separability (Torkkola, 2003).

### 4.1.2 Wrapper Approach vs. Filter Approach

Optimal feature selection and dimensionality reduction coupled with a specific classifier topology, namely the wrapper approach, which maximizes the classification accuracy and minimizes Bayes classification error, results in a combinatorial computational requirement; thus, is unsuitable for adaptive learning of feature projections. On the contrary, the filter approach, which selects features by optimizing certain criterion is independent of the classifier; hence is more flexible. In our baseline system discussed in Chapter 3, we used a committee of three classifiers followed by majority vote. Using a wrapper approach in our baseline system requires additional computation because the classifiers need to be re-trained when using a different subset of features. What's more, if in the future we replace one classifier to another one, we have to repeat the feature selection and dimensionality reduction procedure. Therefore, filter approach is more suitable to our system.

### 4.1.3 Feature Selection and Dimensionality Reduction Criterion - Mutual Information (MI)

In the filter approach, it is important to optimize a criterion that is relevant to Bayes risk, which is typically measured by the probability of error. Therefore, a suitable criterion is mutual information (MI) between the features $\boldsymbol{f}$ and the class labels $c$ as defined by

$$I_S(f;c) = H_S(f) - \sum_c p_c H_S(f \mid c) \qquad \text{(Eq. 4-1)}$$

where $P_c$ is the class prior, $H_S$ and $I_S$ denote Shannon's definitions of entropy and mutual information (Cover and Thomas, 1991). The justification for utilizing mutual information as the optimality criterion for feature selection is that the selected features should contain maximal information about the cognitive states (class labels). This choice of the criterion is also motivated by the fact in information theory that the lower

and upper bounds of the error $p_e$ are related to the mutual information between features and class labels (Fano, 1961; Hellman and Raviv, 1970). For example, in (Hellman and Raviv, 1970) it is shown that $p_e(\boldsymbol{f}) \leq (H_S(c) - I_S(\boldsymbol{f};c))/2$. In principle, MI measures non-linear dependencies between a set of random variables taking into account higher order statistical structures in the data, as opposed to linear and second-order statistical measures such as correlation and covariance.

Since MI is used as the criterion for feature selection and dimensionality, estimating mutual information between features and class labels is critical. We introduce our mutual information estimation method in the next section.

## 4.2 Mutual Information Estimation

Several MI based methods have been developed for feature selection (Battiti, 1994; Ai-ani and Deriche, 2001; Kwak and Choi, 2002; Yang and Moody, 1999; Yang and Moody, 2000). Estimating MI requires the knowledge of the joint pdf of the data in feature space. Evaluating the MI between two scalar random variables (one being the discrete class labels) using histograms has been studied in the literature (Battiti, 1994; Yang and Moody, 2000). However, this approach fails when dealing with high dimensional variables. Torkkola (2003) proposed an approach using a quadratic divergence measure to find an optimal transformation that maximizes the MI between features and class labels. This approach depends on Parzen's density estimation, and therefore is inefficient for subspace projections from high dimensionalities due to the joint density estimation requirement.

A shortcoming of existing MI-based feature selection methods is that, since features are generally mutually dependent, feature selection in this manner is typically suboptimal in the sense of the maximum

joint mutual information principle. In practice, the mutual information must be estimated nonparametrically from the training samples (Hild *et al.*, 2001). Although this is a challenging problem for multiple continuous-valued random variables, the class labels are discrete-valued in the feature transformation setting, which reduces the problem to just estimating entropies of continuous random vectors. Furthermore, if the components of the random vector are independent, the joint entropy becomes the sum of marginal entropies. Thus, the joint mutual information of a feature vector with the class labels is equal to the sum of marginal mutual information of each individual feature with the class labels, providing that the features are independent. In this section, we will introduce a new MI estimation method that combines linear independent component analysis (ICA) with sample-spacing based entropy estimation (Learned-Miller and Fisher III, 2003). We call this method ICA-MI mutual information estimation method. Figure 4-1 illustrates the block diagram of linear ICA-MI framework.

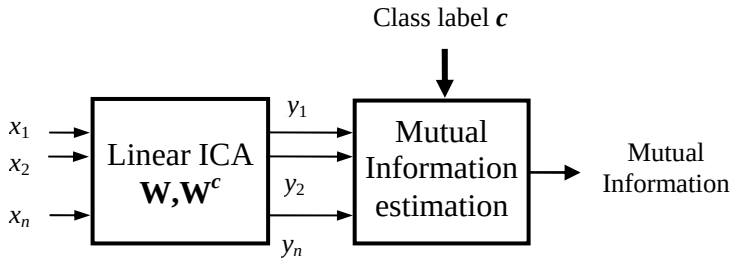



Figure 4-1 Block diagram of linear ICA-MI framework

### 4.2.1 ICA-MI Mutual Information Estimation Framework

This method uses linear independent component analysis (ICA) to separate mixed features into independent features, then conveniently employs single-dimension entropy estimator to estimate mutual information. In

ICA transformation $\mathbf{y}=\mathbf{W}^T\mathbf{x}$, it is easy to show that the relationship between the entropy of the original features $\mathbf{x}$ and the entropy of the projected features $\mathbf{y}$ satisfies (Cover and Thomas, 1991):

$$\begin{aligned} H_S(\mathbf{x}) &= H_S(\mathbf{y}) - \log|\mathbf{W}| \\ H_S(\mathbf{x}\,|\,c) &= H_S(\mathbf{y}\,|\,c) - \log|\mathbf{W}^c| \end{aligned} \qquad \text{(Eq. 4-2)}$$

where $\mathbf{W}$ is the ICA separation matrix for all data, and $\mathbf{W}^c$ is the ICA separation matrix for the data from class $c$. If the components of the random vector $\mathbf{y}$ in (Eq. 4-2) are approximately independent, the joint entropy becomes the sum of marginal entropies. Similarly, if $\mathbf{y}$ conditioned on $c$ has approximately independent components, the conditional joint entropy becomes the sum of marginal-conditional entropies.

$$\begin{aligned} H_S(\mathbf{x}) &= \sum_{l=1}^{n} H_S(y_l) - \log|\mathbf{W}| - I_S(\mathbf{y}) \\ H_S(\mathbf{x}\,|\,c) &= \sum_{l=1}^{n} H_S(y_l) - \log|\mathbf{W}^c| - I_S(\mathbf{y}\,|\,c) \end{aligned} \qquad \text{(Eq. 4-3)}$$

Above, $I_S(\mathbf{y})$ and $I_S(\mathbf{y}|c)$ denote any residual mutual information after the linear ICA procedure, n is the dimension of the features. Assuming that these residual dependencies are negligible, we have

$$\begin{aligned} I_S(\mathbf{x};c) &= H_S(\mathbf{x}) - \textstyle\sum_c p_c H_S(\mathbf{x}\,|\,c) \\ &\approx \sum_{l=1}^{n} \left( H_S(y_l) - \textstyle\sum_c p_c H_S(y_l) \right) \\ &\quad - \left( \log|\mathbf{W}| - \textstyle\sum_c p_c \log|\mathbf{W}^c| \right) \end{aligned} \qquad \text{(Eq. 4-4)}$$

For simplicity, in the following, we further assume that the linear transformations satisfy $\mathbf{W}=\mathbf{W}^c$ for all $c$. Thus,

$$I_S(\mathbf{x};c) \approx I_S(\mathbf{y};c) = \sum_{l=1}^{n} I_S(y_l;c) \qquad \text{(Eq. 4-5)}$$

Consequently, the MI between the classes and resulted new set of statistically independent features can then be computed by evaluating multiple individual MI computations as shown in (Eq. 4-5). We implement this approach by combining Independent Component Analysis transformations with a sample-spacing

based entropy estimator (Learned-Miller and Fisher III, 2003). The implementation details are discussed in the next section.

### 4.2.2 Linear ICA Solution

Given an arbitrary raw feature vector **x**, one can always find a nonlinear transformation **y=f(x)** that is invertible and results in independent components $\mathbf{y}=\{y_1,\ldots,y_n\}$ (Hyvarinen and Pajunen, 1999). However, in situations involving small datasets, finding a robust Nonlinear Independent Component Analysis (NICA) solution is difficult without a priori information about the data distribution. In many practical cases, an approximate linear ICA solution of the form $\mathbf{y}=\mathbf{W}^T\mathbf{x}$ can be sufficient.

There are several efficient algorithms for solving the linear ICA problem based on a variety of assumptions including maximization of non-Gaussianity, minimization of mutual information, nonstationarity of the sources, etc. (Learned-Miller and Fisher III, 2003; Hild *et al*., 2001; Hyvärinen and Oja, 1997). The fourth-order statistical methods can be compactly formulated in the form of a generalized eigendecomposition problem that gives the ICA solution in an analytical form (Parra and Sajda, 2003). Therefore, this formulation will be employed in this paper. According to this formula, the separation matrix **W** is the solution to the following generalized eigendecomposition problem:

$$\mathbf{R_x W} = \mathbf{Q_x W \Lambda} \qquad \text{(Eq. 4-6)}$$

where $\mathbf{R_x}$ is the covariance matrix of **x** and $\mathbf{Q_x}$ is the cumulant matrix estimated using sample averages: $\mathbf{Q_x}=E[\mathbf{x^T x x x^T}]-\mathbf{R_x}tr(\mathbf{R_x})-E[\mathbf{x x^T}]E[\mathbf{x x^T}]-\mathbf{R_x R_x}$. Given the estimates for these matrices, the ICA solution can

be easily determined using efficient generalized eigendecomposition algorithms (or using the *eig* command in Matlab).[1]

Once the ICA transform is determined and employed to obtain **y** such that (Eq. 4-5) holds (even if approximately), the marginal mutual information of each independent feature $y_i$ with the class label $c$ can be computed using (Eq. 4-1) and a simple marginal entropy estimator. One needs to estimate the overall feature entropy $H_S(y_i)$ using all samples regardless of class labels, and the conditional entropy of each class using only the samples from the corresponding class.

### 4.2.3 Marginal Entropy Estimator

There are many entropy estimators in the literature for single-dimensional variables (Beirlant *et al*., 1997). Here, we use sample-spacings estimator, which are based on order statistics. This estimator is selected because of its consistency, rapid asymptotic convergence, and its computational efficiency. Given a set of independent identical distribution (iid) samples of a random variable $Y$ $\{y_1,\ldots,y_N\}$, the estimator first sorts the samples in increasing order such that $y_{(1)}\leq\ldots\leq y_{(N)}$. The $m$-spacing entropy estimator is given in terms of the sorted samples by:

$$\hat{H}(Y) = \frac{1}{N-m}\sum_{i=1}^{N-m}\log\frac{(N+1)(y_{(i+m)}-y_{(i)})}{m} \qquad \text{(Eq. 4-7)}$$

where $N$ is the number of samples, and $m$ is the spacing. This estimator is based on two assumptions: the true density $p(y)$ is approximated by a piecewise uniform density determined by $m$-neighbors; and outside

[1] Note that fourth-order cumulant-based ICA algorithms typically require a much larger sample size than information theoretic methods such as Infomax (Parra and Sajda, 2003) and Mermaid (Beirlant *et al*., 1997), thus has much larger estimation variance for a given sample size. However, the eigenvector formulation is extremely convenient.

of the sample range, the contribution of the true density is negligible and/or does not change the expected entropy computed by (Eq. 4-7).

The selection of the parameter $m$ is determined by a bias-variance trade-off and is typically set so that $m = \sqrt{N}$ . In general, for asymptotic consistency the sequence $m(N)$ should satisfy

$$\lim_{N\to\infty} m(N) = \infty \qquad \lim_{N\to\infty} m(N)/N = 0 \tag{Eq. 4-8}$$

## 4.3 Feature Selection

Feature selection is the procedure which selects a subset of the features from the original feature space and maximizes the classification accuracy given the maximum dimension $d$. From (Eq.4-1), the definition of feature selection can be viewed as finding an optimal subset of features that maximizes the MI between input features and class labels:

$$\max_{\{i_1,\ldots,i_d\}} I(x_{i_1},\ldots,x_{i_d};c) \tag{Eq. 4-9}$$

where $\mathbf{x}$ is the feature vector, it contains all the PSD features from channel 1 to channel $d$, $c$ is the class label, and $d$ is the number of features being considered. Features $\mathbf{x}$ are generally mutually dependent, so $I(\mathbf{x};c)$ can be calculated using our ICA-MI approach in Section 4.2.

Ideally, to find the optimal subset of $d$ features, an exhaustive search or brute-force search is desired. This is a computational complexity of $2^n$. In practice, greedy search, which only involves a complexity of $n(n+1)$, has very similar performance to exhaustive search; hence is widely used. In this paper, we use the greedy search strategy. It first finds one optimal feature; then adds a second feature to find a subset with two optimal features. For any given optimal subset of features $\boldsymbol{x}_d$, we can always pick another one from the

remaining features, and construct the new optimal subset of features $\boldsymbol{x}_{d+1}$. Repeat this procedure, until all features are ranked.

### 4.3.1 EEG Channel Selection

EEG channel selection is to select a subset of EEG electrodes, and can be treated as a feature selection problem. However, unlike usual feature selection, it is necessary to consider all features coming from a channel together (refer to the feature extraction section in Chapter 3), because each EEG channel may contain more than one feature (i.e. different frequency bands of activity in our baseline system). Using the MI estimation method and the greedy search feature selection strategy described in this chapter (also called ICA-MI feature selection method), we can modify the ICA-MI method to make it suitable for EEG channel selection. Table 4-1 shows the salient EEG channel ranking algorithm:

Table 4-1 Salient EEG Channel Ranking Algorithm (Greedy search)

| |
|---|
| A. Estimate the MI between all features in one EEG channel and class labels. Repeat this process for all channels, finding the channel with maximum MI, and mark it as opt-sub1 (optimal subset of 1 channel). |
| B. Pick one from the remaining EEG channels, combine it with opt-sub1 to form sub2 (subset of 2 channels). Estimate MI between all features in sub2 and class labels. Repeat this process for all remaining channels, find the channel with maximum MI, and mark it as opt-sub2. |
| C. Repeat Step B by increasing one channel at a time, until all EEG channels are ranked in the sense of MI maximization. |

This procedure results in an ordering of EEG channels such that the first *d* channels have maximal MI with class labels (approximations include linear ICA induces independence and no degenerate feature pairs

such as the XOR problem exist). The choice of $d$ to be used in the application is dependent on the requirement for classification performance and computational cost.

### 4.3.2 Bias Analysis

The approximations in Section 4.2.1 introduce an estimation bias to each MI evaluation step. From the derivation we can see that the bias, defined as the expected difference between the estimation and the true MI, is simply given by

$$E[\hat{I}_S(\mathbf{x};c) - I_S(\mathbf{x};c)] = \left(\log|\mathbf{W}| - \sum_c p_c \log|\mathbf{W}^c|\right) + \left(I_S(\mathbf{y}) - \sum_c p_c I_S(\mathbf{y}|c)\right) \quad \text{(Eq. 4-10)}$$

where $\mathbf{y}=\mathbf{W}\mathbf{x}$ is the ICA transformation.

## 4.4 Dimensionality Reduction

Even after EEG channel selection, further dimensionality reduction might be desirable to improve classifier generalization performance. This can also be achieved using the ICA-MI framework because an invertible transformation does not change the mutual information. In particular, the linear, invertible ICA mapping yields $I_S(\mathbf{x};c)=I_S(\mathbf{y};c)$. Furthermore, since (Eq. 4-5) holds for the independent features and since MI is a nonnegative quantity, the best $d$-dimensional linear projection consists of the $d$ components of $\mathbf{y}$ that have maximum individual mutual information with $c$. After the ICA mapping, one needs to evaluate the mutual information $I_S(y_i;c)$ for $i=1,\ldots,n$. $n$ is the dimension of the projected features $\mathbf{y}$. The projection matrix then consists of the $d$ columns of the ICA matrix $\mathbf{W}$, that corresponds to the top $d$ components of $\mathbf{y}$. This projection scheme is illustrated in Figure 4-2. Typically, the channel selection procedure described in Section 4.3.1 is employed for selecting the useful sensors motivated by physical constraints. The feature

projection procedure described here is employed for improving classifier robustness and generalization capability motivated by the availability of only a relatively small training data set and the well-known *curse-of-dimensionality*.

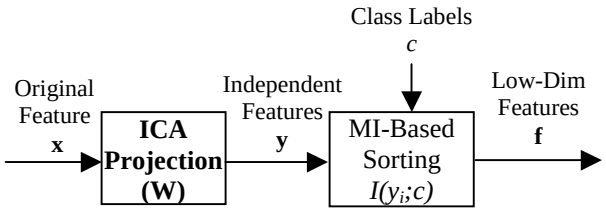



Figure 4-2 Feature projections using ICA preprocessing and mutual information sorting.

## 4.5 Experimental Results

In chapter 3, we applied our baseline system on the AugCog data. Although we collected data using 32 channels, we only used 7 for data processing and classification in our baseline system due to the limitation of the band width and the realtime requirement. The 7 used channels were selected based on the literature. In this section, we will select EEG channels using the method we described in Section 4.3.1, and compared the classification accuracy with that showed in last chapter.

### 4.5.1 EEG Channel Selection

The data collection, preprocessing, feature extraction and classification are exactly the same as in Chapter 3. Three subjects performed two mental tasks at the two designated difficulty levels in this experiment. We first applied the approach on individual subject-task combinations, and obtained specialized EEG channel rankings, designated as *Local n* (*n* is the number of the selected EEG channels). To examine the ability to select optimal channels for all tasks and all subjects, we also used data from all subjects and tasks to get a

new ranking, called *Global n*. An instance of *Local 10* and *Global 10* EEG channels are shown in Table 4-2. The 7 channels used in Chapter 3 are also listed for reference as *Phy 7*. Note that the individual best channels vary for each subject and task combination as expected. Nevertheless, the global ranking strongly coincides with these individual rankings as observed from Table 4-2. This indicates our feature selection method has the ability to find common critical EEG channels among different sessions and subjects, hence has the ability to partly solve session-to-session transfer and subject-to-subject transfer problems.

Table 4-2 Optimal EEG channels illustration. Phy 7: 7 EEG channels from physiological experience; Local 10: 10 best EEG channels evaluated from individual data file; Global 10: 10 best EEG channels evaluated from all data files.

| | | | |
|---|---|---|---|
| Phy 7 | | | Cz, P3, P4, Pz, O2, PO4, F7 |
| Local 10 | $S_1$ | Larson | CP5, Fp2, FC5, Fp1, C4, P4, F7, AF3, P7, FC6 |
| | | n-back | AF3, FC5, Fp1, Fp2, F8, F7, FC6, O1, CP6, P4 |
| | $S_2$ | Larson | Fp2, O1, AF4, F7, C3, PO3, FC6, CP2, C4, Pz |
| | | n-back | C4, O1, F8, Fz, F3, FC5, FC1, C3, Cz, CP1 |
| | $S_3$ | Larson | Fp2, F8, F7, FC5, FC6, AF3, C3, F4, P4, AF4 |
| | | n-back | CP5, F8, C4, FC6, Fp2, FC5, P3, AF4, C3, P7 |
| Global 10 | | | Fp2, FC5, O1, F3, FC6, F8, F7, AF3, O2, CP6 |

To validate the proposed method, we employed a committee of 3 classifiers: GMM, KNN, and Parzen, with majority vote and decision fusion on the selected EEG channels (Refer to chapter 3). For jackknife evaluation of performance, the data for each case is partitioned to five sets and each set is saved for testing using the other four for training. The confusion matrices are estimated and the correct classification rates are calculated. The classification accuracies averaged over the five test sets are shown in Table 4-3. Note that the MI-selected channels significantly outperform the literature-motivated channels. On average,

keeping 7 or 10 channels does not seem to make much difference in accuracy. The MI-selected features perform around 80% accuracy on average for subjects 1, 2, and 3; the local selections are observed to similar to the global selections. The results confirm that the proposed channel selection method can partly solve the subject-to-subject transfer, session-to-session transfer nonstationarity problems. We also observe that in some cases, increasing the number of channels from 7 to 10 increases error, which indicate feature selection can improve the generalization performance, make the system robust.

Table 4-3 Classification rate for three subjects: $S_1$, $S_2$ and $S_3$, in two mental tasks: Larson and n-back, for different subsets of EEG channels. Average is arithmetic average of the 6 classification rate for a particular EEG channel subset.

| | | Phy 7 | 7 Local | 10 Local | 7 Global | 10 Global |
|---|---|---|---|---|---|---|
| $S_1$ | Larson | 0.74 | 0.90 | 0.86 | 0.84 | 0.78 |
| | n-back | 0.78 | 0.89 | 0.87 | 0.85 | 0.83 |
| $S_2$ | Larson | 0.64 | 0.80 | 0.80 | 0.75 | 0.77 |
| | n-back | 0.73 | 0.89 | 0.89 | 0.88 | 0.86 |
| $S_3$ | Larson | 0.48 | 0.71 | 0.76 | 0.76 | 0.73 |
| | n-back | 0.55 | 0.75 | 0.80 | 0.80 | 0.78 |
| Average | | 0.65 | 0.82 | 0.83 | 0.81 | 0.79 |

### 4.5.2 Dimensionality Reduction

By selecting 7 optimal EEG channels, we reduced the dimension to 7X5=35. We could further reduce the dimension by applying the dimensionality reduction method described in Section 4.4. We randomly picked EEG data from one of the subject in one mental task, applied EEG channel selection first as described in 4.5.1, and used the features from local 7 channels as input. Approximately 6000 samples were obtained and used, with 35 dimensional feature vectors (7 selected EEG channels with 5 frequency bands each) and a

desired class label One third of these samples were randomly selected and used as the training set for dimensionality reduction and classification, and the remaining two-thirds samples were used as the testing set. The dimensionality reduction was obtained by the method described in Section 4.4. The correct classification rates for different dimensionality of optimally selected features were evaluated using the classifier committee over 50 Monte Carlo runs (random partitions of training and testing data). The average results are shown in Figure 4-3, from which we see that an accuracy of 80% is achieved with 12 dimensions, while the remaining 23 dimensions do not contribute significantly to the classification accuracy. This indicates using a subset of features, which requires less computational load, can give us similar performance as using the full set of features, hence makes realtime system feasible.

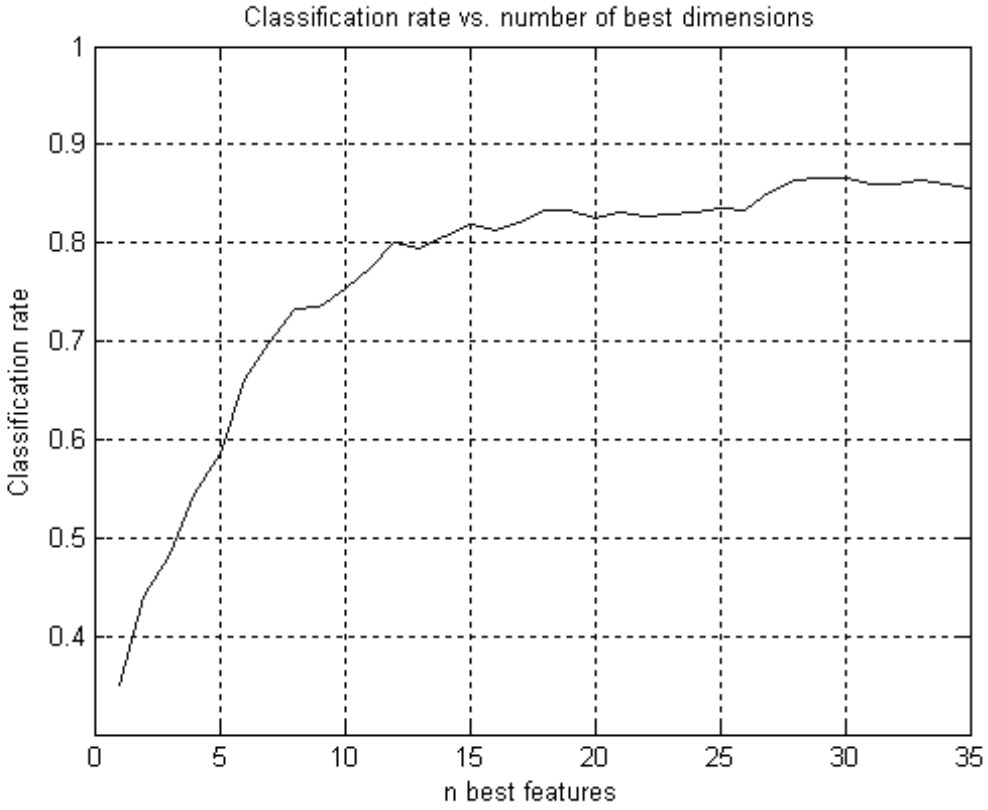



Figure 4-3 Correct classification rate vs. dimensionality of optimally selected features.

## 4.6 Summary

In this chapter we described our ICA-MI framework for estimating mutual information between features and class labels. We then developed our feature selection and dimensionality reduction methods based on this framework. Experimental results showed that our feature selection and dimensionality reduction performed better than our baseline system. Feature selection results also showed that it can partly solved the subject-to-subject transfer, session-to-session transfer non-stationary problem.

In this chapter, we assumed that the linear assumption is satisfied and applied a linear ICA transformation on the EEG data. However, this assumption may not be true, since we have no prior knowledge about the data. In Chapter 5, we describe two non-linear feature selection methods and compare them with linear ICA-MI method described in this chapter.

# Chapter 5: Non-Linear Methods for Feature Selection

In Chapter 4, we introduced linear methods for feature selection and dimensionality reduction in the AugCog system. Experimental results showed that the described feature selection and dimensionality reduction methods can effectively extract salient features and eliminate redundant features, which increased the robustness of the system, and partly solved the session-to-session transfer, subject-to-subject transfer nonstationary problem. However, when the linear assumption does not hold, non-linear methods are more desirable. In the first half of this chapter, we first extend the ICA-MI framework discussed in Section 4.2.1 to the non-linear domain, using piece-wise linear to approximate non-linear solution. Segmenting the data into partitions, we assume the linear assumption holds within each partition and apply the linear ICA-MI framework to the data of each partition. In this way, we generalize our ICA-MI framework into the piece-wise linear domain. The linear ICA-MI framework becomes a special case of this generalized framework, when the number of partitions equals to 1. In the second half of this chapter, another non-linear feature selection method, GMM-MI, is proposed. In this method, we still use MI as the feature selection criterion. Gaussian Mixture Model (GMM) is used to estimate MI between features and class labels. The advantage of this method is that no prior knowledge is required for the distribution of the data. We apply both non-linear feature selection methods on different dataset (not only on EEG data), and present and discuss experimental results. All the work in this chapter was previously published in (Lan and Erdogmus, 2005; Lan *et al*., 2006a; Lan and Erdogmus, 2007; Lan *et al*., 2006b).

## 5.1 Piece-wise Linear Feature Selection Method

As we discussed in chapter 4, the MI based method for feature selection is motivated by lower and upper bounds in information theory (Fano, 1961; Hellman and Raviv, 1970). Fano's and Hellman & Raviv's bounds demonstrate that the probability of error is bounded from below and above by quantities that depend on the Shannon MI between the feature vectors and class labels. Specifically, Hellman & Raviv showed that an upper bound on Bayes error is given by $(H(C)-I(\mathbf{x},C))/2$, where $H(C)$ is the Shannon entropy of the priori probabilities of the classes and $I(\mathbf{x},C)$ is the Shannon MI between the continuous-valued feature vector and the discrete-valued class label. Maximizing this MI reduces the upper bound as well as Fano's lower bound, and therefore, the probability of error decreases.

The linear ICA-MI framework and feature selection algorithm were described in Section 4.2 and 4.3, where a linear ICA projection was applied to separate data into independent components, and then mutual information between features and class labels was estimated by the summation of the marginal entropy. However, if the linearity assumption does not hold, a nonlinear ICA transformation is desirable to achieve independent $\mathbf{y}$. Nonlinear ICA requires more data samples and is computationally intensive. Furthermore, if the transformation is not invertible, the mutual information changes after transformation. So it is not possible to estimate MI using the proposed framework in Figure 4-1.

Karhunen et. al. proposed a local linear ICA algorithm that uses piecewise linear ICA (also called local linear ICA) to approximate nonlinear ICA (Karhunen *et al*., 2000). The idea of local linear ICA is: firstly segment the data into $p$ partitions, then assume that the linearity assumption holds in each partition, and apply linear ICA within each partition. Combining the idea of local linear ICA and ICA-MI framework we introduced in Section 4.1, we can easily extend our linear ICA-MI feature selection method into the nonlinear domain and propose the piece-wise linear ICA-MI framework. What's more, when the number of

partitions equals 1, local linear ICA-MI reduces to linear ICA-MI. We can formulate both linear and nonlinear ICA-MI together and propose the generalized ICA-MI framework.

### 5.1.1 Generalized ICA-MI Mutual Information Estimation Framework

As we mentioned above, linear ICA can be treated as a special case of piece-wise linear ICA when the number of partition equals to 1. We will reformulate Eq.4-2 ~ Eq. 4-10, and combine linear ICA-MI and piece-wise linear ICA-MI into a generalized ICA-MI mutual information estimation framework (Figure 5-1).

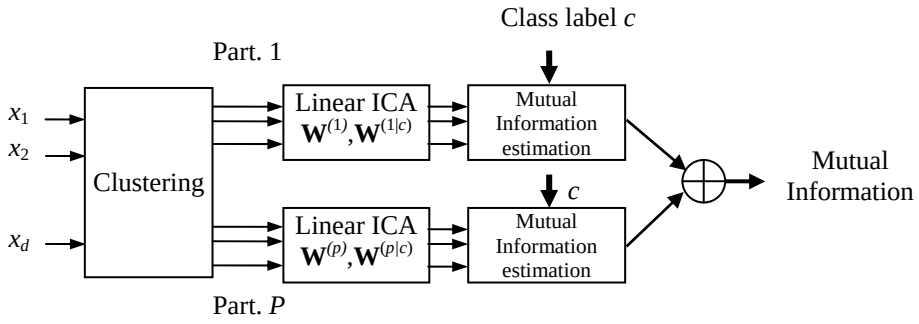



Figure 5-1 Block Diagram of generalized ICA-MI framework for MI estimation

The algorithm is described as follows: First apply a suitable clustering/quantization algorithm to segment the data into $p$ partitions: $\mathbf{x}^{(1)}$, $\mathbf{x}^{(2)}$, …, $\mathbf{x}^{(p)}$; assume that within each partition $\mathbf{x}^{(i)}$, the data is $d$ dimensional (at the $d^{\text{th}}$ step of the ranking procedure, this vector is comprised of previously ranked $d$-1 features and the candidate feature from the unranked ones), and distributed in accordance with the linear ICA model; apply the linear ICA transformation on each partition $C$+1 times (where $C$ is the number of classes. We apply ICA on overall data $\mathbf{x}^{(i)}$ and data from class C, $\mathbf{x}^{(i\backslash c)}$) to get transformed feature vectors for each partition: $\mathbf{y}^{(i|c)}$, which is transformed with class specific linear ICA matrix $\mathbf{W}^{(i|c)}$, and $\mathbf{y}^{(i)}$, which is

transformed with the overall partition ICA matrix $\mathbf{W}^{(i)}$, where *C* denotes class labels. As a result of the linear ICA transformations, we have:

$$\begin{aligned} H(\mathbf{x}^{(i)}) &= H(\mathbf{y}^{(i)}) - \log|\mathbf{W}^{(i)}| \\ H(\mathbf{x}^{(i|c)}) &= H(\mathbf{y}^{(i|c)}) - \log|\mathbf{W}^{(i|c)}| \end{aligned} \quad \text{(Eq. 5-1)}$$

where $i = 1,\ldots,p$. $H(\mathbf{x}^{(i)})$ is the entropy for cluster *(i)*, and $H(\mathbf{x}^{(i|c)})$ is the conditional entropy for cluster *(i)* in class *C*. If linear ICA works perfectly, then the joint entropies of $\mathbf{y}^{(i|c)}$ and $\mathbf{y}^{(i)}$ reduce to the sum of marginal entropies. However, this is not guaranteed, therefore, the residual mutual information will remain as an estimation bias. In practice, we have an imperfect ICA solution and

$$\begin{aligned} H(\mathbf{x}^{(i)}) &= \sum_{l=1}^{d} H(y_l^{(i)}) - \log|\mathbf{W}^{(i)}| - I(\mathbf{y}^{(i)}) \\ H(\mathbf{x}^{(i|c)}) &= \sum_{l=1}^{d} H(y_l^{(i|c)}) - \log|\mathbf{W}^{(i|c)}| - I(\mathbf{y}^{(i|c)}) \end{aligned} \quad \text{(Eq. 5-2)}$$

Where $q_i$ denotes the probability mass of the corresponding partition. Mutual information satisfies the following additivity property for any partition:

$$I(\mathbf{x}; C) = \sum_{i=1}^{p} q_i I(\mathbf{x}^{(i)}, c) \quad \text{(Eq. 5-3)}$$

The mutual information within each partition can be expressed as a linear combination of entropy values as follows:

$$I(\mathbf{x}^{(i)}, C) = H(\mathbf{x}^{(i)}) - \sum_c p_{ic} H(\mathbf{x}^{(i)} | c) \quad \text{(Eq. 5-4)}$$

where $p_{ic}$ denotes the probability mass of class C in partition *i*. Substituting (Eq. 5-2) in (Eq. 5-4)

$$\begin{aligned} I(\mathbf{x}^{(i)}, C) = &\left( \sum_{l=1}^{d} H(y_l^{(i)}) - \sum_c p_{ic} \sum_{l=1}^{d} H(y_l^{(i|c)}) \right) \\ &- \left( \log|\mathbf{W}^{i}| - \sum_c p_{ic} \log|\mathbf{W}^{i|c}| \right) \\ &- \left( I(\mathbf{y}^{(i)}) - \sum_c p_{ic} I(\mathbf{y}^{(i|c)}) \right) \end{aligned} \quad \text{(Eq. 5-5)}$$

The last parenthesis in (Eq.5-5) shows the estimation bias one makes when estimating the MI within each partition, if it is assumed that the local linear ICA solution in that partition achieved perfect separation. Over all partitions, the total estimation bias (estimated MI minus the actual MI) is averaged as follows:

$$Bias = \sum_{i=1}^{p} q_i \left( I(\mathbf{y}^{(i)}) - \sum_{c} p_{ic} I(\mathbf{y}^{(i|c)}) \right) \quad \text{(Eq. 5-6)}$$

Note that as the number of partitions approaches infinity asymptotically, one could utilize a grid partitioning structure within which the probability distributions would be uniform, thus local linear ICA would achieve perfect separation within each infinitesimal hypercube. However, in practice, one cannot utilize infinitely number of partitions given a finite number of samples. Note that the analysis above also holds for the case where linear ICA is employed directly on the whole dataset without any partitions.

The decomposition of mutual information into overlapping segments (cover rather than partition) has been previously studied by Szummer and Jaakkola in the context of model regularization in the presence of unlabeled data and semisupervised learning (Szummer and Jaakkola, 2002). The partition approach we propose here is along the same line of reasoning, that is, the cumulative relevant information of a feature vector can be decomposed to local regions in the vector space. However, while Szummer and Jaakkola are interested in emphasizing discriminative and dense regions in the data for density fitting, we are interested in estimating the total useful information in a feature vector.

### 5.1.2 Mutual Information Estimation under the Generalized ICA-MI Framework

The first step of the MI estimation under generalized ICA-MI framework is that to segment the data into *p* non-overlapping partitions. In theory, for infinitesimal partitions the linearity assumption always holds within partitions; however, small sample size prevents reliable linear ICA estimates. Therefore, the tradeoff

between the number of partitions and samples per partition must be considered in the bias-variance framework. Cross-validation can be used to determine the proper number of partitions.

The data is partitioned using the K-means clustering algorithm (Haykin, 1998). The K-means algorithm tries to minimize the average squared distance of the data to the centers of clusters:

$$J = \sum_{i} \sum_{\mathbf{x}^{(i)} \in S_i} \left\| \mathbf{x}^{(i)} - \mathbf{m}_i \right\|^2 \qquad \text{(Eq. 5-7)}$$

where $\mathbf{m}_i$ is the center of each cluster. This algorithm first selects $K$ random cluster centers, and then calculates the distance between all data points to these clusters centers respectively. Samples are assigned to the cluster corresponding to the nearest center and then cluster centers are updated to the average of the assigned samples. The process is repeated until $J$ converges to its minimum value (local minimum).

After partitioning, linear ICA is solved in each partition using generalized eigendecomposition of 2$^{nd}$ and 4$^{th}$ order cumulant matrices. One-dimensional entropies are estimated using the sample spacing approach. The mutual information between feature vectors and class labels within each partition can be estimated using the same method as described in Section 4.2. The overall mutual information between feature vectors and class labels can be estimated using (Eq. 5-3).

### 5.1.3 Feature Selection Algorithm under Generalized ICA-MI Framework

Feature selection is the process that maximizes mutual information between selected features and class labels, as in (Eq. 4-9). Similar to Section 4.3, we use greedy search strategy to avoid exhaustive search in feature space. The algorithm starts from finding one optimal feature, then adds a second feature, and together with the first one, finds two optimal features. Mutual information is estimated using the method

described in Section 5.1.2. For any given optimal subset of features $\boldsymbol{x}_d$, we can always pick another one from the remaining features, and construct the new optimal subset of features $\boldsymbol{x}_{d+1}$. Repeat this procedure, until all features are ranked.

### 5.1.4 Pilot Experiment on Synthetic Dataset

In order to illustrate the difference between linear ICA-MI feature selection method described in Chapter 4 and piece-wise local ICA-MI feature selection method introduced in this chapter, we apply both methods on a synthetic dataset. This dataset consists of four features: $x_i$ ($i$=1,…,4), where $x_1$ and $x_2$ are nonlinearly related (Figure 5-2 left), $x_3$ and $x_4$ are independent from the first two features and are linearly correlated Gaussian-distributed with different mean and variance (Figure 5-2 right). There are two classes in this dataset (represented as different grayscale levels in print). These two classes are separable in the $x_1$ and $x_2$ plane, but overlapping in the $x_3$ and $x_4$ plane. It is clear that this dataset can be well classified only using $x_1$ and $x_2$, while $x_3$ and $x_4$ provide redundant and insufficient information for perfect classification. From Figure 5-2 we can see that $x_2$ has less overlap compared with $x_1$, while $x_3$ has less overlap than $x_4$. So ideally, the feature ranking in descending order of importance in terms of classification rate should be $x_2$, $x_1$, $x_3$, $x_4$. In our experiments, we choose the sample size as 1000 and use 20 partitions. The optimal number of partitions can be achieved by M-fold cross-validation process. The '+' in Figure 5-2 represents the partition centers. We also apply linear ICA without any partitioning. We repeat the above experiment for 100 Monte Carlo runs. The linear ICA approach finds the ranking to be $x_4$, $x_3$, $x_2$, $x_1$, while the local linear ICA approach with 20 partitions finds the expected *correct* ranking.

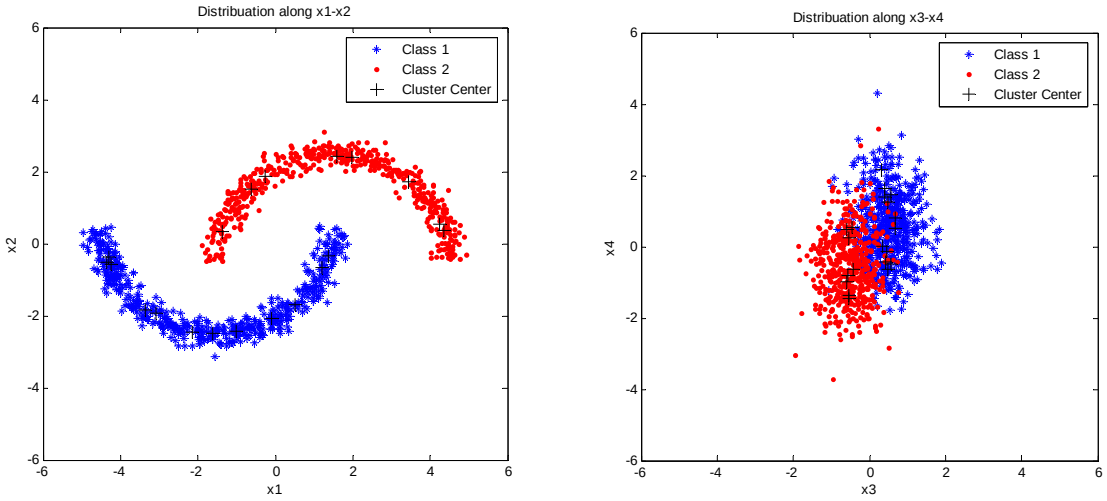



Figure 5-2 Four-dimensional Synthetic dataset and corresponding cluster centers. Left: distribution of $x_1$ and $x_2$; Right: distribution of $x_3$ and $x_4$.

### 5.1.5 Experiment on UCI Iris Dataset

In this experiment, we apply linear ICA-MI and piece-wise linear ICA-MI (with 2 partitions) feature selection methods to the ranking of the features for the Iris dataset from the UCI database (http://www.ics.uci.edu/~mlearn/MLRepository.html). Due to the small sample size, 10 Monte Carlo rankings with randomly selected training (used for ranking) and test sets are utilized, each consisting of 50% of the available samples. For each ranked subset, a Gaussian Mixture Model (GMM) based Bayesian classifier is employed. The frequency of rankings and classification accuracy are shown in Table 5-1 and Figure 5-3. Since both methods agree on $x_4$ as the top one, pairwise scatter plots of this feature with the remaining features are shown in Figure 5-4 for visual comparison. $x_3$ seems to yield a more compact class distribution, while $x_1$ or $x_2$ and $x_4$ seem to have less overlapping samples. Still, it is difficult to judge and we rely on the GMM performances on the testing set for the final comparison. The classification accuracy in Figure 5-3 shows that local linear ICA yields more accurate feature ranking than linear ICA in Iris data.

Table 5-1 Feature ranking frequencies on the Iris dataset.

| Methods | Ranking indices |
|---|---|
| Linear ICA-MI | 4 3 2 1 |
| Local linear ICA-MI | 4 1 2 3 |
| | 4 2 3 1 |
| | 4 2 1 3 |

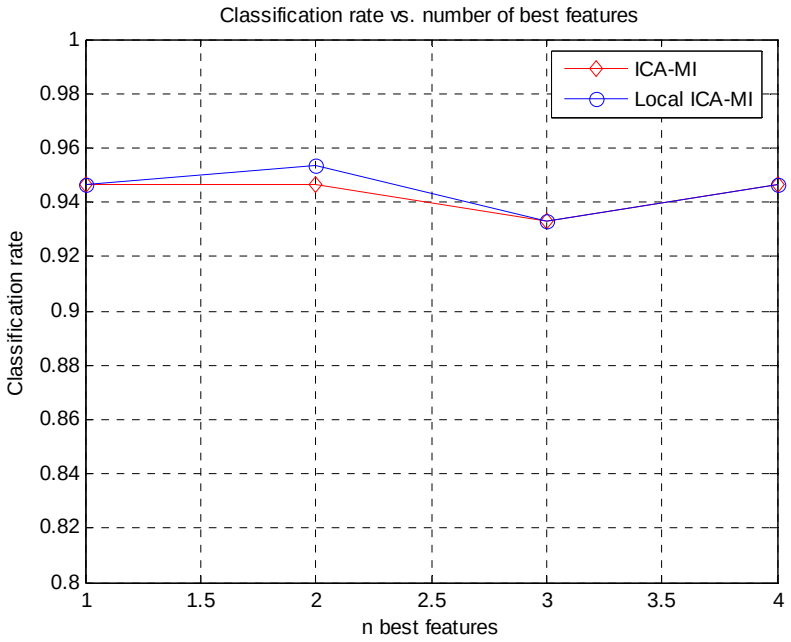



Figure 5-3 Classification accuracy for Iris data by linear ICA-MI and Local linear ICA-MI methods. The classification accuracy is the average over 10 Monte Carlo simulations.

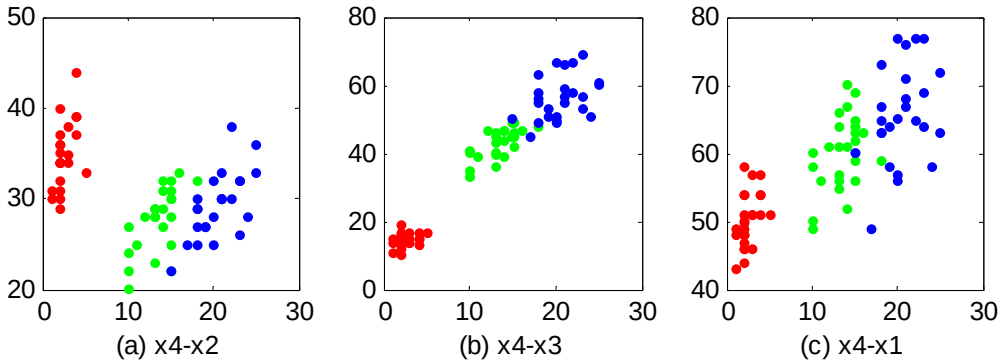



Figure 5-4 Combinational distribution of 2 feature vectors of Iris dataset. Left: distribution of $x_4$ and $x_2$; Middle: distribution of $x_4$ and $x_3$; Right: distribution of $x_4$ and $x_1$.

### 5.1.6 Experiments on AugCog Dataset

In Chapter 3, we applied out baseline system without using feature (EEG channel) selection and dimensionality. In Chapter 4, we applied linear ICA-MI feature selection and dimensionality method on the same dataset based on linear assumption. In this section, we will apply piece-wise linear ICA-MI feature selection method and compare the results with those in Chapter 4.

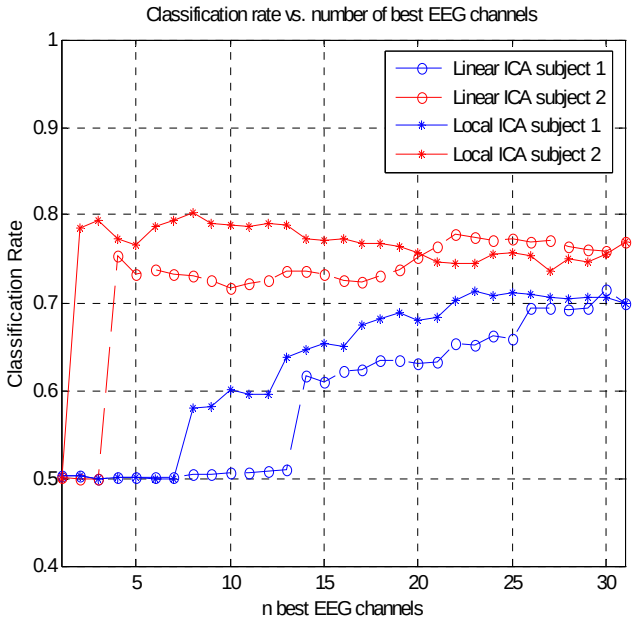



Figure 5-5 EEG channel ranking in terms of classification rate for two subjects by linear ICA and local linear ICA.

The data collection, preprocessing, feature extraction and classification are exactly the same as in Chapter 3. The EEG channel selection results evaluated by correct classification rate are shown in Figure 5-5 (when a channel is selected/discarded all five features associated with the channel are selected/discarded). For comparison, we also illustrate the performance using linear ICA-MI for EEG channel ranking on both subjects. The solid line with stars illustrates the classification results for local ICA, while the dashed line with circles illustrates the classification results for linear ICA for both subjects. We observe that local ICA

outperforms linear ICA in both subjects. EEG channels ranked based on to both methods are shown in Table 5-2. Both methods pick the same first 3 features for subject 1, but pick totally different features for subject 2. This indicate EEG based BCIs are highly data dependable.

To compare feature ranking/selection results for linear ICA and local ICA more clearly, we list the EEG channel ranking in descending order in terms of contribution to classification for both subjects in the Table 5-2. For both subjects, linear and local ICA-MI methods give different EEG channel ranking results.

Table 5-2 EEG channel ranking (descending order) in terms of contribution to classification rate for subject 1 and subject 2 with linear ICA and local ICA methods.

| Subject | Method | EEG channel ranking |
|---|---|---|
| Sub 1 | Linear ICA-MI | FC2, AF3, CPZ, FP1, CP5, CP1, C4, CP6, P3, CP2, F4, F3, PO4, O2, P4, O1, PZ, P8, FCZ, FC1, FC6, AF4, FC5, FZ, P7, F8, CZ, FP2, F7, PO3, OZ |
| | Local linear ICA-MI | FC2, AF3, CPZ, AF4, FC5, F7, CZ, O2, F3, F4, FC6, C4, F8, P3, FP2, CP6, P8, PZ, P7, FZ, FC1, OZ, PO3, FCZ, FP1, CP2, CP1, P4, CP5, PO4, O1 |
| Sub 2 | Linear ICA-MI | FC1, CP1, CZ, O1, C4, F3, FCZ, FC2, FZ, CP2, AF3, FP1, CP6, F4, P3, CPZ, CP5, AF4, FC6, P7, PO4, OZ, PZ, PO3, P4, F8, FC5, O2, F7, FP2, P8 |
| | Local linear ICA-MI | CP1, O1, FP1, CZ, 'FC1, P8, PO4, FP2, FCZ, P7, F4, P3, P4, PO3, CP6, FC6, CPZ, FC5, AF4, FZ, F3, CP5, F7, F8, AF3, CP2, C4, PZ, FC2, O2, OZ |

## 5.2 Gaussian Mixture Model Feature Selection Method

In the first half of this chapter, we described a non-linear feature selection that used piece-wise linear ICA-MI framework to estimate mutual information between feature vectors and class labels. Experimental results showed that when the linear assumption did not hold, piece-wise linear feature selection method outperformed the linear feature selection method described in Chapter 4. However, from (Eq. 5.6) we know that the accuracy of mutual information estimation relies on the ICA performance within each partition.

Ideally, if a partition is limited within a small area and has infinite data samples, the projected features, which ICA finds are nearly independent; hence the mutual information estimation is accurate. However, in real world application, we always have limited data samples with high dimensions, in which case a cross validation process is used to find the optimal number of partitions. Where we have a sparse number of samples compared to the number of dimensions, further dividing data into partitions can only decrease the performance of ICA transformation. In the second half of this chapter, we exploit the fact that Gaussian Mixture Model (GMM) is widely used for density estimation for an arbitrary distribution and can be used for mutual information estimation, and propose another non-linear feature selection method: GMM-MI.

### 5.2.1 GMM-MI Mutual Information Estimation Framework

Similar to the ICA-MI framework we described in Chapter 4, the goal of GMM-MI framework is also to estimate mutual information between feature vectors and class labels, but using Gaussian Mixture Model. We revisit Eq. 4-1 and rewrite it as:

$$I_S(x;c) = H_S(x) - \sum_c p_c H_S(x \mid c) \qquad \text{(Eq. 5-8)}$$

where $I_s(x;c)$ is the shannon MI between feature vectors **x** and $c$ , which is defined in terms of the entropies of the overall data and individual classes. $p_c$ are the prior class probabilities. The entropy is given by

$$\begin{aligned} H_S(x) &= -\int p(x)\log\, p(x)dx \\ H_S(x \mid c) &= -\int p(x \mid c)\log\, p(x \mid c)dx \end{aligned} \qquad \text{(Eq. 5-9)}$$

where $p(\mathbf{x}|c)$ are the class conditional distributions. The overall data distribution is

$$p(x) = \sum_c p_c\, p(x \mid c) \qquad \text{(Eq. 5-10)}$$

The above equations show that the critical step for feature selection is the entropy estimation. Previously in the ICA-MI framework, we estimated entropy by an indirect way. In the GMM-MI framework, since one

can approximate an arbitrary distribution by a limited number of Gaussian components with sufficient amount of data, one can estimate entropy directly by definition Eq.5-9. The GMM density estimation can be written as:

$$p(x) = \sum_{m=1}^{M} \alpha_m G_m(x) \quad \text{(Eq. 5-11)}$$

where $G_m(x)$ is the distribution of each Gaussian component, and $\alpha_m$ is the corresponding component prior. So the estimation of overall entropy is:

$$\hat{H}_S(x) = -\frac{1}{N}\sum_{i=1}^{N} \log\left(\sum_{m=1}^{M} \alpha_m G_m(x_i)\right) \quad \text{(Eq. 5-12)}$$

where the class conditional entropy is given by:

$$\hat{H}_S(x \mid c) = -\frac{1}{N_c}\sum_{i=1}^{N_c} \log\left(\sum_{m=1}^{Mc} \alpha_m G_m(x_i^c)\right) \quad \text{(Eq. 5-13)}$$

where N is the overall data sample, $N_C$ is the data sample for class *C*, and $x^c$ is the data sample from class *C*. Combining (Eq. 5-8), (Eq. 5-12) and (Eq. 5-13), the MI estimation can be written as:

$$\begin{aligned} I(x;c) = &-\frac{1}{N}\sum_{i=1}^{N} \log\left(\sum_{m=1}^{M} \alpha_m G_m(x_i)\right) \\ &+ \sum_c p_c\left(\frac{1}{N_c}\sum_{i=1}^{N_c} \log\left(\sum_{m=1}^{M_c} \alpha_m G_m(x_i^c)\right)\right) \end{aligned} \quad \text{(Eq. 5-14)}$$

In the next section, we describe a new non-linear feature selection method, that is based on the GMM-MI framework.

### 5.2.2 Feature Selection Algorithm using GMM-MI Framework

Using the mutual information estimation given by (Eq. 5-14), with greedy search strategy, we can easily develop GMM-MI based feature selection algorithm. However, every time we estimate mutual information

with different subset of feature vectors, we need to: 1) apply cross-validation to determine the optimal number of components; and 2) fit the data using the Expectation-Maximization algorithm (Dempster *et al.*, 1977). Thus, feature selection using this procedure can be time-consuming, and is not practical in real-world applications. To overcome this difficulty, we first use spherical covariance matrix, and assume that the number of optimal components for all features is identical to that for different combinations of feature subsets. In this way, we only need to do cross-validation once for all features at the beginning, and pick rows and columns from the mean vectors and covariance matrices for the corresponding features. Under this assumption, the GMM-MI feature selection algorithm can be written as shown in Table 5-3.

Table 5-3 GMM-MI feature selection algorithm

| |
|---|
| A. Use cross-validation to determine the optimal number of components for each class and overall data for all feature vectors. Estimate both class densities and overall density for all feature vectors using GMM. Get the mean vectors and covariance matrices for each component. |
| B. Pick the corresponding rows and columns from mean vectors and covariance matrices (generated in A), and estimate density for each feature vector. Estimate the MI between each feature vector and class labels. Find the feature with maximum MI, and mark it as opt-sub1 (optimal subset of 1 feature). |
| C. Pick one of the remaining feature vectors, combine it with opt-sub1 to form sub2 (subset of 2 features). Find the corresponding rows and columns from mean vectors and covariance matrices (generated in A), form the new mean vectors and covariance matrices, and estimate both class densities and overall density of sub2, and then estimate MI between sub2 and class labels. Repeat this process for all remaining features, find the features with maximum MI, and mark it as opt-sub2. |
| D. Repeat Step C by increasing one feature at a time, until all features are ranked in the sense of MI maximization. |

### 5.2.3 Experiments on AugCog Dataset

We apply the GMM-MI feature selection method on AugCog dataset used in the previous chapters. The EEG data collection, preprocessing, feature extraction are exactly the same as in Chapter 3. The experimental procedure is similar to that of described in Section 5.1.6, except for two differences: 1) We mix data from different subjects and different sessions; and 2) to simplify the procedure, instead of using the committee of 3 classifiers described in Chapter 3, we only use a GMM classifier. The feature selection results are evaluated by classification accuracy for both linear ICA-MI and GMM-MI methods, and are shown in Figure 5-6. The ranked EEG channel indices and average computation time are also listed in Table 5-4:

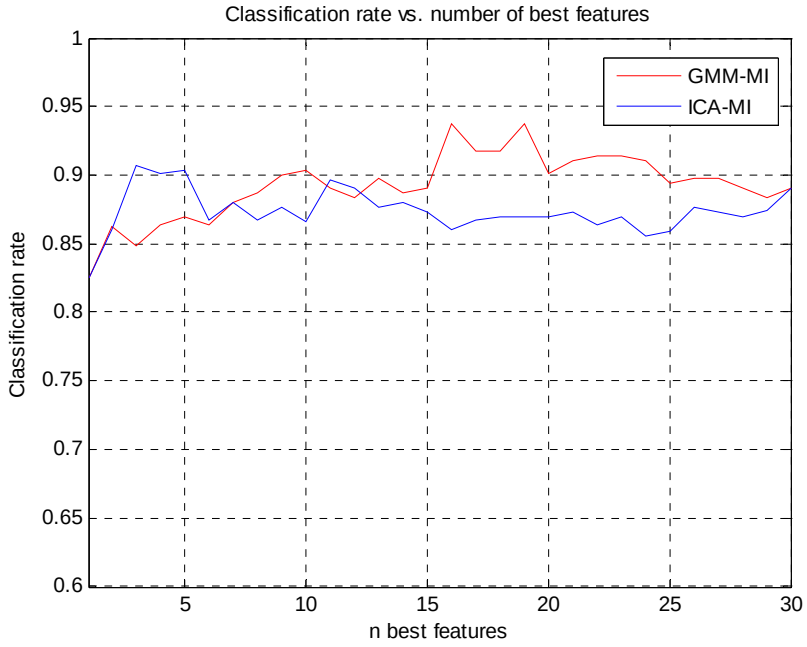



Figure 5-6 Classification accuracy for EEG data by GMM-MI and ICA-MI algorithms.

The experimental results show that ICA-MI method is much faster than GMM-Mi method. From the performance point of view, both GMM-MI and ICA-MI exhibit certain degree of consistency. However,

none of them is superior to the other: if we only select 3-5 features, ICA-MI yields better performance; if we want to select 15-20 features, GMM-MI yields better performance. For the ICA-MI algorithm, as we mentioned before, linear assumption might degrade the accuracy of MI estimation. For GMM-MI algorithm, there could be two reasons: 1) the assumption we used in Section 5.2.2, that the optimal numbers of components are identical for different combination of features, might not hold in some cases; and 2) we do not have enough data.

Table 5-4 Ranked EEG channel indices and computation time for both feature selection methods.

| | Average CPU time (second) | Ranking indices |
|---|---|---|
| GMM-MI | 558.86 | 24 3 23 7 11 8 30 2 4 20 26 9 6 1 29 17 16 10 5 18 14 12 19 21 28 25 27 15 13 22 |
| ICA-MI | 2.86 | 24 18 3 22 13 23 29 7 16 28 2 5 14 27 25 6 1 10 11 19 8 20 4 30 9 15 21 17 12 26 |

## 5.3 Discussion

In this chapter, we described two non-linear feature selection methods: the generalized ICA-MI method and the GMM-MI method. The former one uses piece-wise linear to approximate linear ICA transformation for mutual information estimation, while the latter one uses Gaussian Mixture Model density estimation to estimate mutual information. Both methods work when the linear assumption described in Chapter 4 does not hold.

Different results from synthetic dataset, UCI iris dataset, and AugCog dataset showed that piece-wise linear ICA-MI feature selection method outperformed linear ICA-MI method. Another advantage of piece-wise linear ICA-MI method is that it can be treated as a generalized framework which contains linear ICA-MI framework as its special case. However, this approximation requires that we have enough data samples

within each partition. Otherwise the projection error using the piece-wise linear ICA can be larger than that of linear ICA.

A second non-linear feature selection method, GMM-MI, is also described in the second half of this chapter in case we don't have enough data samples for each partition. GMM-MI method exploits the fact that GMM is a non-parametric method for density estimation and requires no prior knowledge about the data. The GMM-MI mutual information estimation framework is introduced, and a simplified GMM-MI feature selection algorithm is described in Section 5.2.2. Experimental results on AugCog dataset do not show that GMM-MI performs better than linear ICA-MI. The GMM-MI method is also much slower than the ICA-MI method. However, when linear assumption does not hold, and we do not have enough data samples to apply piece-wise linear ICA-MI method, GMM-MI offers another option for feature selection.

We already explored both linear and non-linear feature selection methods that use mutual information as criterion in the last two chapters. Experimental results showed that the classification performance can be improved to some extent compared to the baseline system introduced in Chapter 3.

# Chapter 6: Statistical Similarity based Feature Extraction Method

In the baseline system described Chapter 3, we used the Power Spectrum Density (PSD) of the EEG signal as features. The PSD features are estimated using a sliding window at 1Hz resolution, and are integrated over pre-defined frequency bands, such as delta, theta, alpha, beta, and gamma. These frequency bands are based on prior experimental and clinical EEG-based studies (Gevins *et al*., 1997). However, these frequency bands can not be guaranteed to yield features that are optimal for a specific application, because various complex mental tasks may involve contributing factors across different frequency bands, or exhibit different characteristics with these frequency bands. In this situation, an adaptive approach is more desirable to determine the importance and correlation among different frequency components. In Chapters 4 and 5, we used the same feature extraction method as in Chapter 3, but focused our studies on different feature selection and dimensionality reduction methods. In this chapter, we revisit the feature extraction component illustrated in Figure 3-2 and introduce an adaptive feature extraction method using statistical similarity. This chapter is based on publication (Lan *et al*., 2006c).

## 6.1 Background

The Power spectrum density (PSD) of EEG signals is a widely used feature for Brain Computer Interfaces. Many researchers have studied how to select frequency components from EEG signals. Pregenzer and Pfurtscheller (1999) used Distinctive Sensitive Learning Vector Quantization to analyze and rank 40 integer frequency components from 2 EEG channels. We developed a mutual information maximization approach for feature ranking and EEG channel selection based on PSD features (Lan *et al*., 2005b).

However, few studies have been performed on how to adaptively and optimally segment the EEG activity into coherent frequency bands.

Due to the standard windowing technique for PSD estimation and smoothness of the expected PSD activity across frequency, it is expected that the optimal task-relevant frequency bands will consist of compactly connected neighboring frequency intervals (determined by the frequency resolution of the PSD). The beginning and endpoint of each band, however, must be determined adaptively by investigating the statistical similarity between the frequency intervals that will be potentially integrated in the same frequency band. Note that intuitively, we should cluster together only the frequency intervals that carry statistically similar information. This will increase the signal-to-noise ratio of the generated feature after integration of PSD over the band and will potentially improve the classification accuracy. Consequently, we first need to measure how statistically similar two frequency intervals are. The simplest such measure in statistics is the correlation coefficient between them. For the rest of this chapter, we will assume that the frequency resolution is 1Hz, thus each frequency interval will be represented by the integer frequency value that the interval is centered at. If two integer frequency components are highly correlated, then they carry redundant information and the combined information is little more than each individual frequency. Thus, combining these correlated frequency components together (for example by averaging, which is what integration does essentially) instead of using them as separate features, will reduce feature dimensionality without sacrificing significant amounts of novel information. In the finite training data case, the generalization benefits one would obtain through reduction in dimensionality will typically surpass the losses incurred due to eliminated information. A more general measure of similarity would be mutual information between pairs of frequencies. However, upon inspection, we have determined that the pairwise

mutual information (estimated nonparametrically using kernel density estimation) and correlation coefficient matrices for our particular experimental datasets revealed similar clustering structures. Therefore, for the rest of this chapter, we focus on correlation coefficients as the primary similarity measure, and adaptively identify the most coherent frequency bands by clustering. Details of the mathematical motivation for the use of correlation coefficient will be described in the next section.

## 6.2 Method

In this section, we will describe the statistical similarity based feature extraction method in detail. The block diagram of the proposed method is shown in Figure 6-1. Recall the AugCog system in Figure 3-2, after preprocessing, including filtering and artifact removal, we get *clean* multi-channel EEG signals. The integer PSD frequency components are estimated using the Welch method (Welch, 1967) from 1 to 40 Hz. We measure the distance between pairwise frequency components using the correlation coefficient matrix. Given the PSD $E(f)$ at each integer frequency $f$ from 1 to 40 Hz, we construct the frequency-correlation matrix **C**:

$$\mathbf{C} = \begin{bmatrix} \cdots & \cdots & \cdots \\ \cdots & C(i,\, j) & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \qquad \text{(Eq. 6-1)}$$

Each element of **C** is the absolute correlation coefficient between $E(i)$ and $E(j)$:

$$C(i,\, j) = \frac{|\, E[(E(i) - \mu_i)(E(j) - \mu_j)]\,|}{Std[E(i)] \cdot Std[E(j)]} \qquad \text{(Eq. 6-2)}$$

where $\mu_i$ denotes $E[E(i)]$. If the correlations between pairwise frequency components are very strong (close to 1) or very weak (close to 0), then the correlation matrix **C** approximately becomes block-diagonal. A typical correlation matrix for one EEG channel is shown in Figure 6-2.

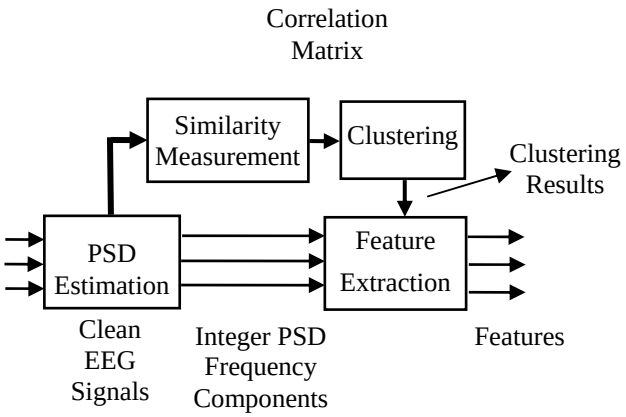



Figure 6-1 Block diagram of statistical similarity based feature extraction method

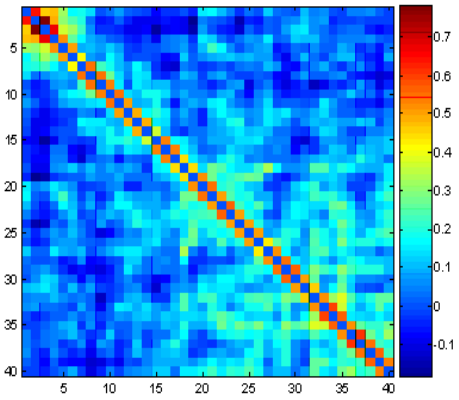



Figure 6-2 A typical correlation matrix for one EEG channel

Once we obtain the correlation matrix, we can employ similarity-based clustering algorithms on this matrix to automatically segment the frequency bands, which we describe next.

### 6.2.1 Spectrum Clustering

Many existing spectral clustering algorithms can achieve this goal (Zha *et al.*, 2001; Hagen and Kahng, 1992; Belkin and Niyogi, 2002; Bach and Jordan, 2003). Spectral clustering algorithms in the literature typically deal with similarity matrices formed between pairs of data samples, and for large data sets with $N$ samples, the similarity matrix becomes $N \times N$. Note that the matrix is essentially a fully connected weighted

graph between the nodes (the samples in spectral clustering or frequencies in our case). Therefore, the procedure of cutting the weakest connection and then searching for the remaining connected components is not feasible for very large *N*. In our application, however, the size of the correlation matrix, determined by the frequency resolution of the PSD estimator, is quite small; therefore, we opt for this straightforward procedure and employ the well known connected component search algorithm (Cormen *et al*., 1990).

### 6.2.2 Frequency Component Integration

Suppose that, after clustering, we get a group of frequencies — $f_1$, $f_2$, …, $f_l$ — which have strong correlations. Following the typical assumption of a linear generative model for the EEG measurements at each electrode, we can consider specific frequencies as corresponding to a certain brain signature, a common source denoted as *g*. Consequently, this underlying common feature is assumed to take various realizations at each frequency as follows:

$$g(f_i) = g + n(f_i) \qquad \text{(Eq. 6-3)}$$

where $n(f_i)$ is the background and measurement noise. Note that this is a simplified linear model and more elaborate linear or nonlinear generative mechanisms will be assumed and tested in future work. Given the model in (Eq. 6-3), the common source is extracted by an appropriate weighted average scheme:

$$g \approx \sum_{i=1}^{l} w_i g(f_i) \qquad \text{(Eq. 6-4)}$$

to maximize classification performance. For simplicity, we selected $w_i=1/l$ and observed this value to work well in practice. In general, optimization of these parameters could be necessary. Based on the model in (Eq. 6-3), we can see that the weighted average feature *g* improves the signal-to-noise ratio, and thus results in a better feature.

### 6.2.3 Algorithm

Based on the previous discussion, the statistical similarity based feature extraction method is summarized in Table 6-1:

Table 6-1 Statistical similarity based feature extraction algorithm

1. Estimate PSD at integer frequencies from artifact-free EEG.
2. For each EEG channel, calculate the correlation matrix using (Eq. 6-1) and (Eq. 6-2).
3. For each channel, find a threshold such that when all entries of **C** below the threshold are zeroed, the connected components algorithm yields a predetermined *K* number of clusters.
4. Integrate signal power in each frequency band as in (Eq. 6-4) determined for each channel to obtain the reduced feature set.

## 6.3 Experimental Results

To compare the statistical similarity based feature extraction method with the one described in Chapter 3, we apply the feature method described in Section 6.2 to the AugCog dataset. The EEG signals are preprocessed and the PSD was estimated from 1 to 40 Hz using the procedure described in Chapter 3. The frequency resolution of the PSD is 1 Hz.

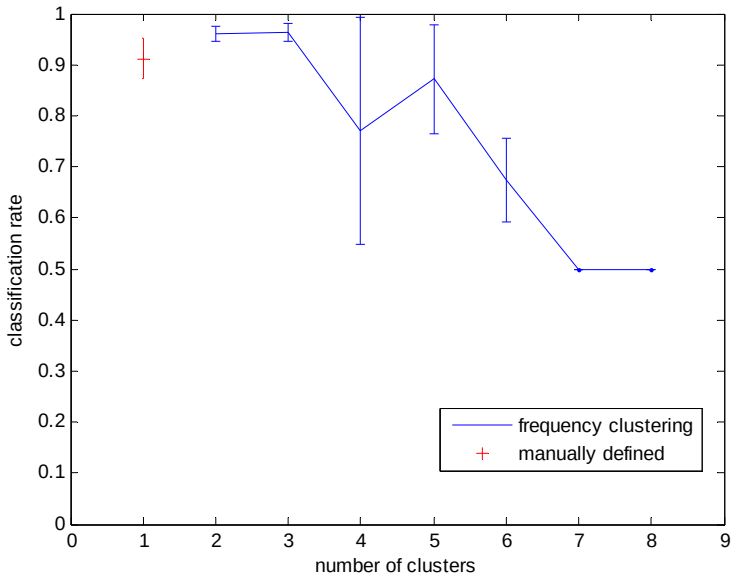
The optimal number of frequency clusters for each EEG channel is acquired by a cross-validation procedure. Data from each session is partitioned into 5 pieces for 5-fold cross-validation. We employ a Gaussian Mixture Model (GMM) based classifier that generates cognitive load estimates at 10Hz. These estimates are passed through a 2-second-long causal median filter to eliminate occasional outliers and to obtain a smooth cognitive state estimation sequence. The average and standard deviation of correct classification probability is used as the cross-validation measure for order selection in identifying the number of frequency bands.

Ideally, one should employ cross-validation to select the number of clusters for each EEG channel, as well as the model order for the GMM classifier. For a $C$-channel EEG recording if we evaluate $K$ different frequency cluster models (that is for each channel evaluate the performance of 1 to $K$ clusters) and $M$ different GMM orders (1 to $M$ Gaussian components), the computational complexity of the cross-validation becomes $K^C MN$. The factor $N$ is the number of random initializations of the EM algorithm to find the global optimum for GMM training. As this complexity tends to increase quite fast, we simplify the search by assuming a predetermined order ($M$=4) for the GMM classifiers, based on our previous experience with datasets collected using this equipment in similar experimental setups. We also assume that each EEG channel uses the same number of frequency bands, thus the power-dependency in $C$ is eliminated. The computational complexity then reduces to $KN$ 5-fold cross-validation procedures. The overall experimental procedure is shown in Table 6-2:

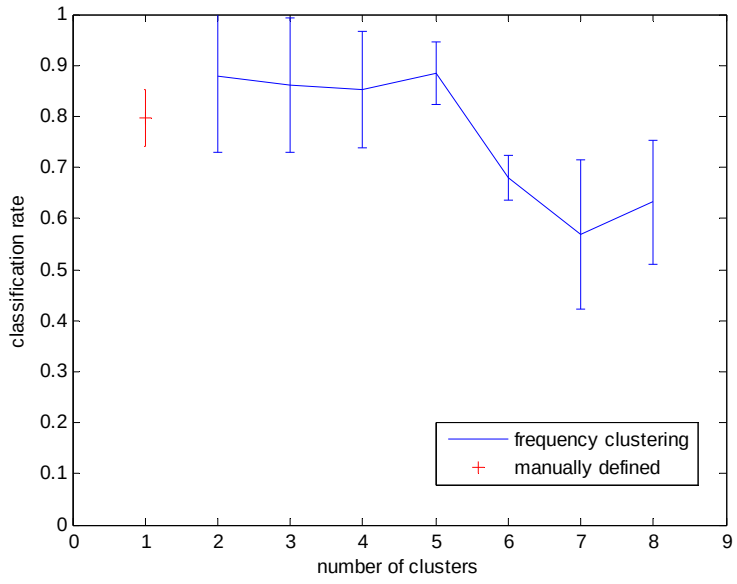
Table 6-2 Experiment procedure for statistical similarity based feature extraction method

1. For each session of data, select a random 5-fold partition.
2. Pick 4 for training (TRAIN) and 1 for testing (TEST)
3. For $K$ from 2 to 8 perform the following:
   - Obtain the reduced dimension features corresponding to the $K$ clusters determined by the segmentation algorithm.
   - On the reduced dimension features, train 100 randomly initialized GMM classifiers.
   - Pick the GMM that maximizes the classification performance on TRAIN.
4. Go to step 2 and repeat until all partitions are used as TEST.
5. Calculate the average and standard deviation of classification error on TEST for the 5 partitions for each K using the best GMMs.
6. Repeat steps 2 to 5 for each session.

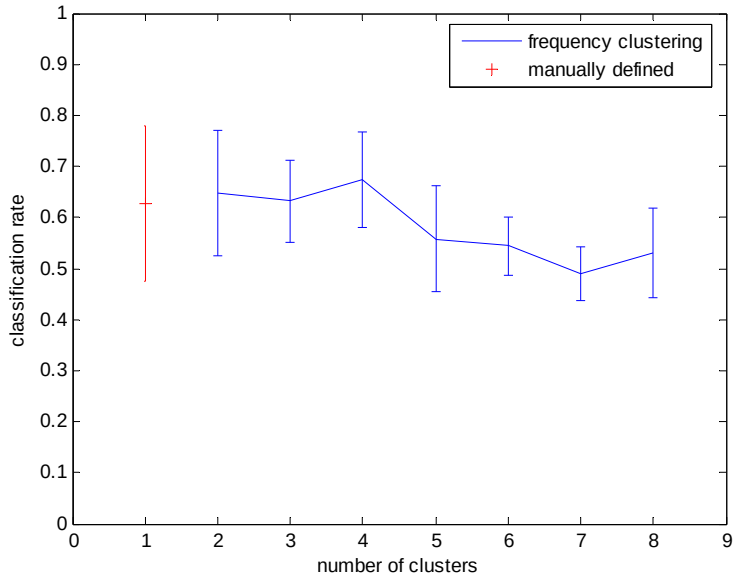
The features obtained by automatic segmentation of frequency bands are compared to features obtained by selecting 5 frequency bands in accordance with the clinical and cognitive science literature. These predetermined frequency bands are 4-8Hz, 8-12Hz, 12-16Hz, 16-30Hz, 30-44Hz. The PSD features are extracted by integrating over these frequency bands, which generates 5 features for each EEG channel. The classifiers are trained on these features using Monte Carlo initialization and the best performing classifiers are selected in the 5-fold cross-validation scheme. The mean and standard deviation of correct classification rate for each subject for different number of clusters per channel are listed in Table 6-3. The performance comparison between automatic frequency clustering and manually predefined frequency bands is shown in Figure 6-3. Experimental results show that for the given dataset, the statistical similarity based feature extraction method outperforms the previously used predefined frequency bands for all subjects given that we use the optimal number of clusters. However, this optimal number is subject and data dependent, which have to be determined using cross-validation.




(a) Subject 1




(b) Subject 2




(c) Subject 3

Fig. 6-3. Performance comparison between the automatic frequency band segmentation (for 2 to 8 bands) and the manually predefined frequency bands in terms of correct classification probability (between 2 classes). The latter is depicted as a separate error bar located at x-axis value of 1.

Table 6-3 Mean and standard deviation of classification rate for different subjects. The second column presents the results for manually predefined frequency bands; columns 3 to 9 correspond to *K* equals 2 to 8 automatically determined frequency bands. The results are shown in the form of mean+/-std.

| | | Number of clusters per EEG channels | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Manual | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Subject 1 | 0.91+/-0.04 | 0.96+/-0.01 | **0.96+/-0.02** | 0.77+/-0.22 | 0.87+/-0.11 | 0.67+/-0.08 | 0.5+/-0 | 0.5+/-0 |
| Subject 2 | 0.80+/-0.06 | 0.88+/-0.15 | 0.86+/-0.13 | 0.85+/-0.11 | **0.88+/-0.06** | 0.68+/-0.04 | 0.57+/-0.15 | 0.63+/-0.12 |
| Subject 3 | 0.63+/-0.15 | 0.65+/-0.12 | 0.63+/-0.08 | **0.67+/-0.09** | 0.56+/-0.10 | 0.54+/-0.06 | 0.49+/-0.05 | 0.53+/-0.09 |

## 6.4 Conclusion and Discussion

In this chapter, we introduced an adaptive feature extraction method that is based on statistical similarity. Compared with the feature extraction method, that integrates frequency components using pre-defined frequency bands, which is used throughout Chapter 3 to Chapter 5, the proposed method effectively discover the correlation within different components. The integration process eliminates the redundant information and increases the Signal-to-Noise ratio. Experimental results showed that the method introduced in this chapter outperformed the previously used feature extraction method.

In Chapters 3 to 6, we focused on one of the BCI applications, Augmented Cognition. In Chapter 3, we described a baseline AugCog system that used the signal processing and machine learning techniques to estimate mental states of human subjects from EEG data when subjects executed mental tasks. In Chapter 4 and Chapter 5, we focused our study on linear and non-linear feature selection and dimensionality reduction method that used mutual information between feature vectors and class labels as criterion. Experimental results showed that by carefully selecting feature selection and dimensionality reduction method, the classification accuracy of the AugCog system can be improved. Furthermore, selecting a

compact and informative set of features also partly solves the session-to-session transfer, subject-to-subject transfer, and non-stationary problem. In the next chapter, we will introduce a second application of the BCI: single trial ERP detection. We first introduce a baseline single trial ERP detection system and then focus on feature extraction, feature selection and dimensionality reduction.

# Chapter 7: Single Trial ERP Detection in RSVP Paradigm

In the first half of this thesis, we introduced one of the BCI applications, Augmented Cognition. In this chapter, we will introduce a second BCI application, called single trial ERP detection. An Event-Related Potential (ERP) is a series of peaks of EEG signals when a stimulus occurs. Among all different types of ERPs, P300, which is a positive potential and happens around 300ms after the stimulus appears, has been extensively studied. P300 responses to rare and meaningful stimuli, which is also called the oddball stimuli (Donchin and Coles, 1988; Johnson, 1988). Positive perturbation in EEG appears after around 300ms of the target stimulus (Figure 7-1). Therefore, P300 is suitable for target detection.

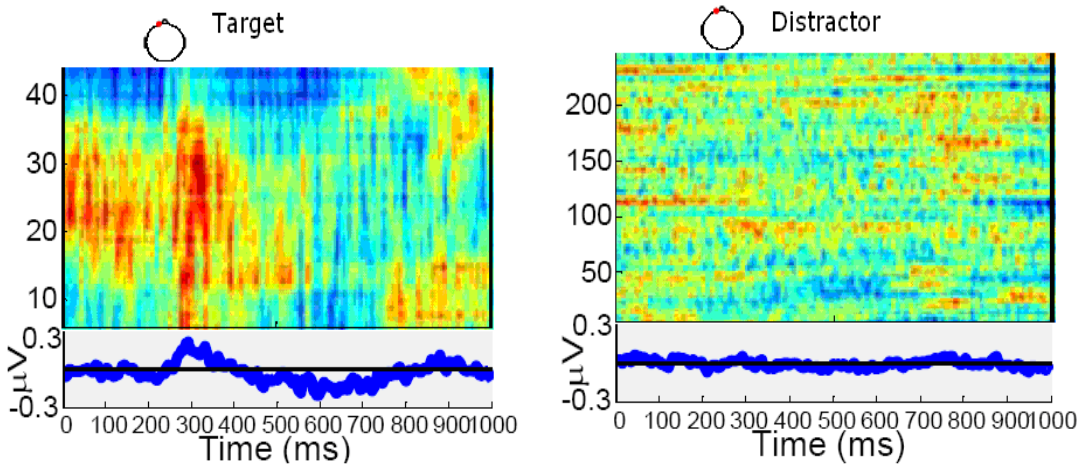



Figure 7-1 ERP vs. non-ERP followed by image stimuli average over trials. The left figure (ERP) is the EEG signals within 1 second window after target image appears. The right figure (non-ERP) is the EEG signals within 1 second window after distractor image appears. In Both figures, the x-axis denotes time, y-axis denotes different EEG channels (Huang *et al*., 2008).

Thorpe's work showed that event related potentials (ERP) can be used for target detection in a Rapid Serial Visual Presentation (RSVP) task (Thorpe *et al*., 1996). In the RSVP paradigm, a series of images, in which only a few of them contain the target images, are presented to a human subject at very fast speed, i.e.

3-20 images/second (Figure 7-2). Although there are different ways to present the stimuli to subjects, we will use RSVP paradigm in our study.

The ERP detection is challenging due to the limited signal-to-noise ratio (SNR) of the non-invasive measurement. Eye blink, facial muscle movement and environment noise can contaminate ERP signals. Conventionally, ERP is studied by averaging stimulus-locked responses from multiple trials. However, the averaging process not only eliminates useful information about brain dynamics, but also compromises bandwidth of communication in a BCI setup. Recently, single trial ERP detection over multi-channel EEG collection received increasing interest due to its numerous potential applications, such as object recognition in brain computer interfaces, and information search from a large image database (Lemm *et al*., 2005; Gerson et al., 2006; Huang *et al*., 2006a, Huang *et al*., 2006b; Sajda *et al*., 2007; Huang *et al*., 2008). Huang et al. investigated several machine learning techniques for single trail ERP detection, including linear and nonlinear detectors (Huang *et al*., 2006a), a boosting algorithm (Huang *et al*., 2006b), and an SVM detector (Huang *et al*., 2008). Among all ERP detectors designed and reported, SVM (with Gaussian kernels) yields the performance, with particular classification accuracy from 75% to 95%, depending on subject and session when 32-channel EEG is utilized. Parra and his colleagues investigated a series of linear algorithms for ERP detection (Parra *et al*., 2005; Parra *et al*., 2008) and reported 92% accuracy across five subjects (to our knowledge, the Parra-Sajda group prefers using 64 electrodes, which makes a few percent addition to performance in our experience).

In this chapter, we will describe a baseline single ERP detection system, which includes experimental setup, data acquisition, signal processing, and ERP detection.

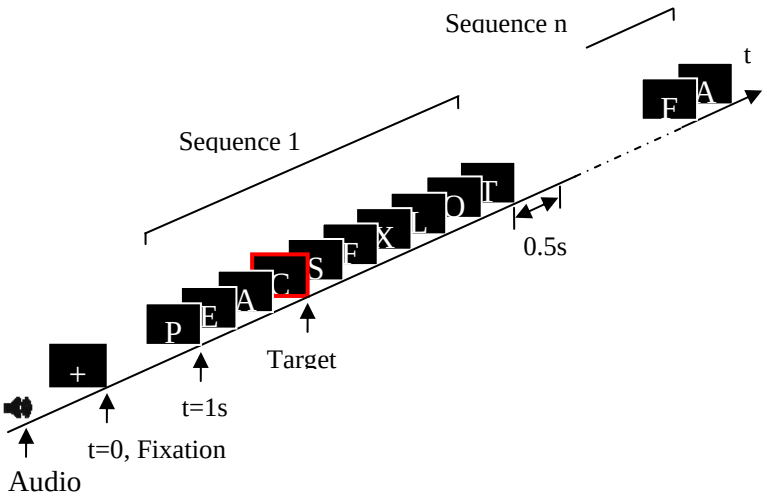



Figure 7-2 An example of a RSVP trial

## 7.1 Data Collection

Subjects were recruited for the study under an approved IRB protocol for RSVP and EEG acquisition. Each subject finished at least two sessions in different days. Each session contained multiple trials. Each trial started with an audio presentation of the target image. At time stamp 0, a one-second fixation screen was presented, followed by *n* sequences of images. Each sequence contained *m* images, at the rate *T* ms/image in a random order. Within each sequence, there was only one target image, all other images were distracters. There was 0.5 second interval between two sequences. An example of an RSVP trial is shown in Figure 7-2.

We used two computers to acquire data, one for image display and the other for data collection. The EEG data were collected using a 32-channel Biosemi ActiveTwo system at sampling rate of 256Hz. Presentation™ (Neurobehavioral Systems, Albany, CA) software was used to present images with a high degree of temporal precision and to output pulses or triggers to mark the onset of the target and each

distracter stimuli. The triggers were received by the Biosemi system over a parallel port and recorded concurrently with the EEG signals.

## 7.2 Signal Processing

We now discuss signal processing for our baseline system in the following section.

### 7.2.1 Pre-processing

The raw EEG signals were filtered using a bandpass filter with a pass band of 1-45Hz. Then the synchronization signals and time stamps were extracted from the data. Since ERPs only happen between 150ms and 450ms after the onset of the stimuli, the filtered data were labeled from the onset of each image stimuli, and truncated using a 600ms window, starting from -100ms of the stimuli onset, and ending after 500ms of the onset.

### 7.2.2 Feature Extraction

The truncated signals were divided into two parts for all EEG channels, from time -100ms to onset of stimuli, called pre-stimulus window, and from onset of the stimuli to 500ms, called stimulus window. The data in stimulus window of each channel were normalized separately using the data from pre-stimulus data. After normalization, data within the stimulus window of all EEG channels were concatenated into one feature vector. An example of feature extraction is shown in Figure 7-3.

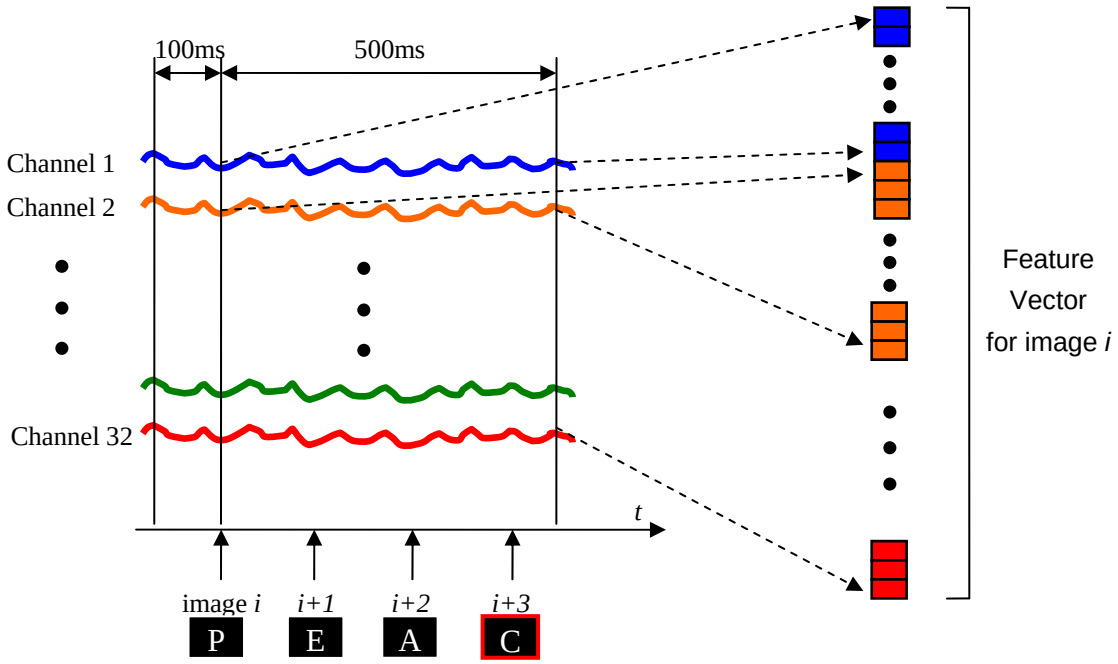



Figure 7-3 An example of feature extraction

## 7.3 ERP Detector

Our goal is to build an ERP detector to accurately detect the ERPs associated with the target stimuli. From Figure 1-2 we know, an ERP detector is virtually a classifier. In our experiment, we found that the SVM classifier and the LDA classifier generally yield better performance than other classifiers (Huang *et al*., 2008; Lan *et al*., 2009; Lan *et al*., 2010a; Lan *et al*., 2010b). Especially in our application, where we have only a few samples with high-dimensional features, most classifiers fail.

### 7.3.1 Support Vector Machine (SVM)

A Support Vector Machine classifies data by projecting the original features into a higher dimensional space and constructing a hyperplane in this space that maximizes the separation. If the projection is linear,

the classifier is called a linear SVM. Most problems can be solved with a linear SVM. However, when the decision function that is used to separate data is not a linear function, a non-linear SVM is needed. In non-linear SVMs, the data is firstly mapped to another space (with infinite dimensions) using the non-linear kernel function $K$, and then a linear SVM is applied on the new feature space.

In our baseline system, we use a non-linear SVM as our ERP detector (Burges, 1998; Duda *et al*., 2000). A radial basis (Gaussian) kernel SVM is used. The kernel size $\sigma$ and the cost parameter $C$ can be set using cross-validation or chosen by the designer. To avoid overfitting, in previous studies we adopted 10-fold cross-validation (Burges, 1998) to adjust these model and regularization parameters. However, this process takes a long time when a brute-force grid-search approach is taken to find the global optimal in a preset domain. Therefore, based on our experience, for our particular data, we have proceeded with the selections $\sigma$=100 and $C$=10.

### 7.3.2 Linear Discriminant Analysis (LDA)

Instead of mapping data into a high dimensional space, fisher LDA projects the data into a 1-D space that maximizes the separability of the data. The projection that LDA looks for is one that maximizes the variance between classes, and to minimize the variance within classes. Due to the ability of reducing dimensions, LDA is also widely used in dimensionality reduction.

Both SVM and LDA classifiers are popular in P300 target detection. Krusienski compared different classifiers and found that the SVM was not necessarily the best classifier in the traditional P300 speller systems (Krusienski *et al*., 2006). Krusienski also found that given a properly selected EEG channel subset, an LDA based classifier outperformed SVM in the target detection accuracy and robustness. More

importantly, compared to the SVM, LDA classifiers require far less computation in the training process, which makes them more suitable for adaptive realtime systems. In Krusienski's study, pre-determined channel subsets were compared and used throughout the study. Though Krusienski claimed the selected subset exhibited session-to-session transfer ability, no explicit results demonstrated that this selection also transferred between subjects. In our experiment, we found that both SVM and LDA could yield better performance, depending on the data.

## 7.4 Experimental Results

Four subjects were recruited for the experiment. Each subject finished eight sessions in four days (one session in the morning, one session in the afternoon). Each session contained 200 trials, each of which lasted 5 seconds. A trial contained one second fixation followed by 40 images (512x512) displayed at 100ms/image. At the same time, we monitored their brain activity via EEG and recorded all data for subsequent analysis. EEG signals were processed offline. We applied the pre-processing, feature extraction method mentioned in section 7.2.2 to get features with 4128 dimensions, then applied the SVM classifier on the feature vectors. We used the data from the morning sessions as training set and the data from afternoon sessions on the same day as testing set. The results are represented using ROC curves, and the area under ROC curves. The ROC curves are shown in Figures 7-4, 7-5, 7-6 and 7-7 for each subject. The area under the ROC curves (AUC) of four days data from all subjects are listed in Table 7-1, and minimum false alarm rate at zero missing (MFAR) of four days data from all subjects are listed in Table 7-2. The average area under ROC curve exceeds 90%. This indicates the baseline system has the ability to detect ERPs. We will discuss the results more in Section 7.5.

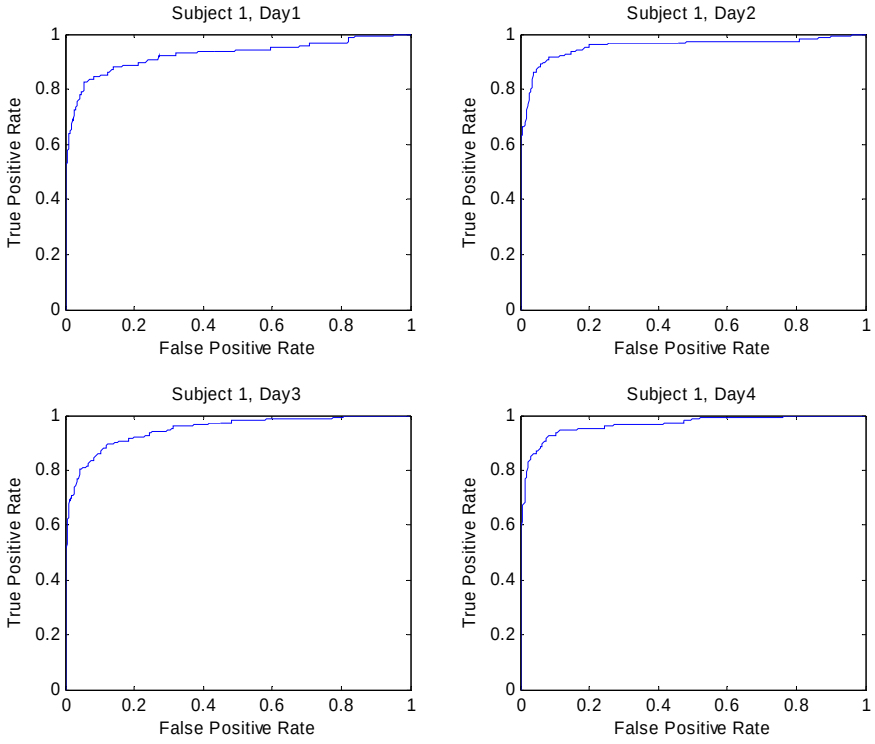



Figure 7-4 ROC curves of four days data from subject 1

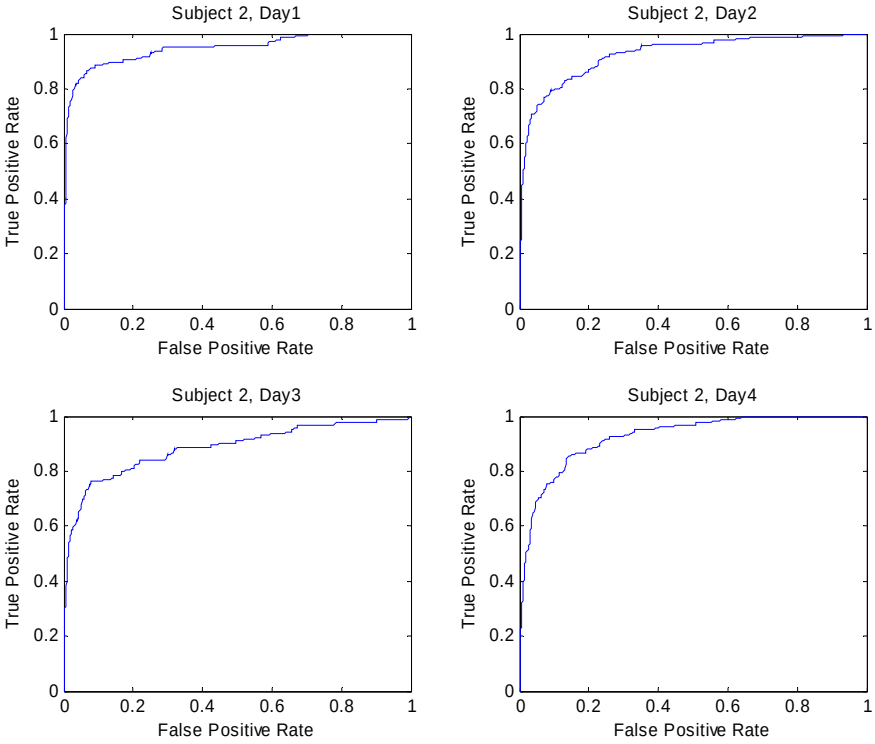



Figure 7-5 ROC curves of four days data from subject 2

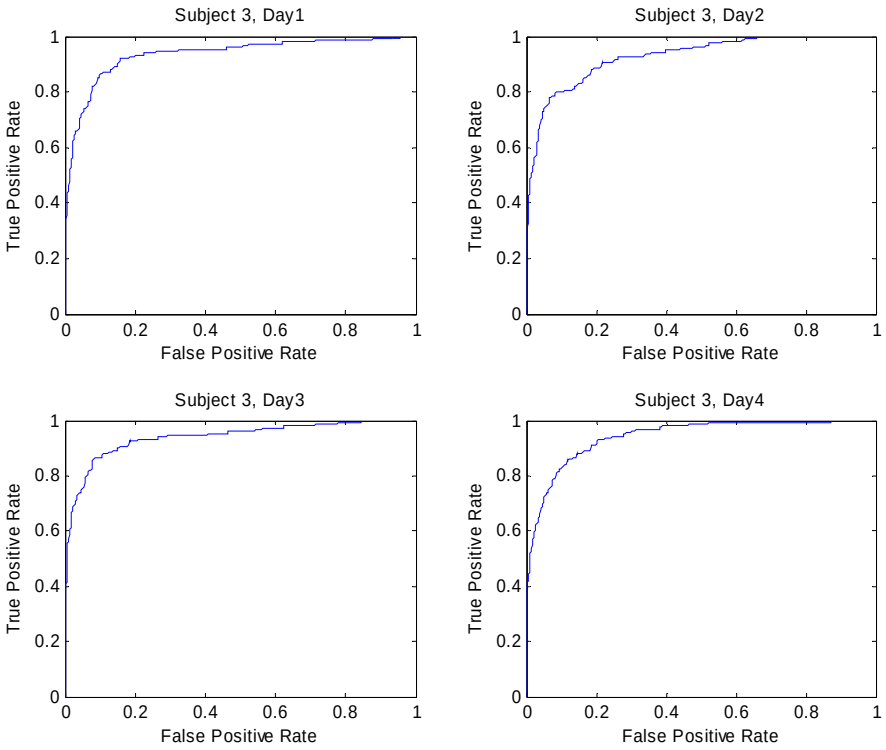



Figure 7-6 ROC curves of four days data from subject 3

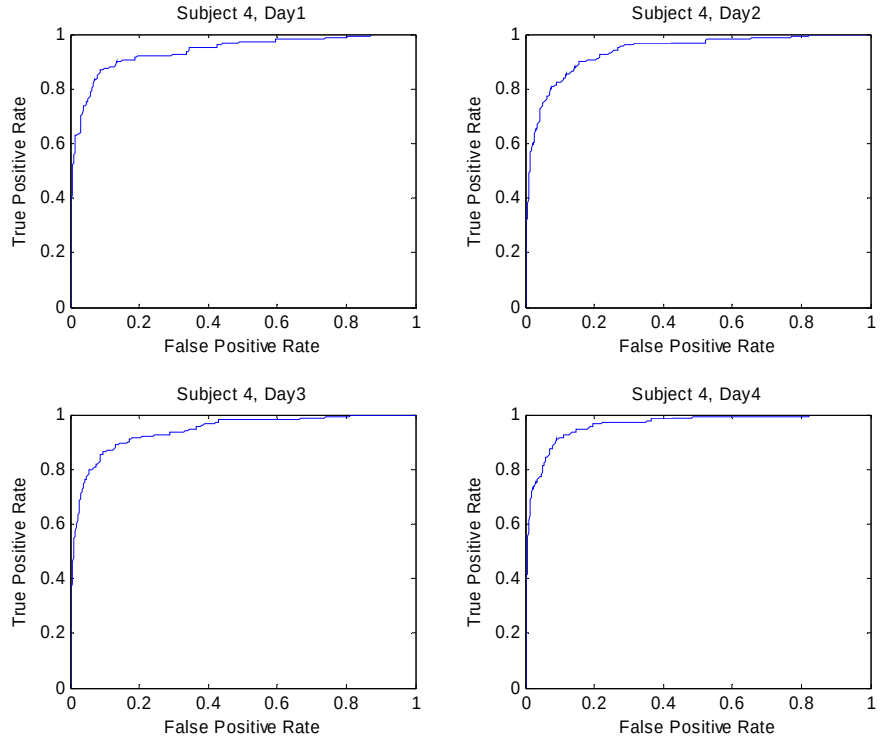



Figure 7-7 ROC curves of four days data from subject 4

Table 7-1 Area under ROC curves of four days data from four subjects

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| Subject 1 | 0.92 | 0.96 | 0.95 | 0.97 |
| Subject 2 | 0.95 | 0.93 | 0.88 | 0.92 |
| Subject 3 | 0.93 | 0.93 | 0.94 | 0.94 |
| Subject 4 | 0.94 | 0.94 | 0.94 | 0.96 |

Table 7-2 MFAR of four days data from four subjects

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| Subject 1 | 0.95 | 0.96 | 0.81 | 0.77 |
| Subject 2 | 0.71 | 0.93 | 1.00 | 0.64 |
| Subject 3 | 0.96 | 0.66 | 0.85 | 0.87 |
| Subject 4 | 0.87 | 0.83 | 0.81 | 0.82 |

## 7.5 Discussion

In this chapter, we introduced a second BCI application, single trial ERP detection. The concept of single trial ERP detection and an example of it under RSVP paradigm were described. A baseline system was proposed to solve the single trail ERP detection problem. Like other EEG based BCI systems (Figure 1-2), this baseline system contained data collection, pre-processing, feature extraction and classifier (ERP detector). Filtered EEG signals were truncated using half-second window from the onset of the stimuli. The truncated signals of all EEG channels were concatenated as a feature vector. For the ERP detector, we experimentally used both SVM and LDA classifiers for their simplicity and good performance.

We applied the baseline system on four day EEG data from four subjects (we used the SVM classifier in this experiment). Experimental results were presented using ROC curves, the area under ROC curves,

and the minimum false alarm rate at zero missing. Usually, in a classification system, the area under ROC curve corresponds to the classification accuracy. The larger the area, the more accurate the classifier is. However, in the single trial ERP detection task, since the data structure is very unbalanced, which contains many more distracters than targets, using the area under ROC curves itself to evaluate the performance of ERP detector is not enough. In some applications when the cost of missing a target is huge, the minimum false alarm rate at zero missing is also used as a criterion. In other applications, a high ERP detection rate with a reasonable false alarm rate is more desirable. From the results shown in Section 7.4, we can see that the overall area under ROC curves exceeded 90%. However, the overall MFAR is big as well.

In the baseline system, we did not use much machine learning techniques to boost the performance. In the next chapter, we will introduce a temporal windowing scheme for the single trial ERP detection.

# Chapter 8: Temporal Windowing Scheme for Single Trial ERP Detection

In Chapter 7, we introduced a baseline system for single trial ERP detection. Time domain features were used for the ERP detection. To extract features, we truncated the filtered data using a 0~500ms window from the onset of the stimuli, then concatenated data for all EEG channels to get feature vectors (Figure 7-3). The extracted features were sent to a SVM classifier for the ERP detection. We applied the baseline system on EEG data from four subjects in four days respectively. Experimental results showed that the overall classification accuracy exceeded 90%. In Section 7.5, we discussed that overall classification accuracy may not be enough to evaluate the performance of an ERP detector, because the data is unbalanced. Other criteria, such as minimum false alarm rate at zero missing or maximum ERP detection rate at low false alarm, are also used besides the area under ROC curve in different applications.

In this chapter, we extend our baseline system by studying different temporal windowing schemes for single trial ERP detection. All the work in this chapter was previously published in (Lan *et al*., 2009).

## 8.1 Background

In Chapter 7, we mentioned that single trial ERP detection is challenging due to the limited Signal-to-Noise ratio (SNR) of the EEG signals. In past years, researchers have developed different machine learning techniques for single trial ERP detection (Huang *et al*., 2006a; Huang *et al*., 2006b; Huang *et al*., 2008). However, based on our experience, single trial ERP detection is highly data dependent and no single method is superior to others. Parra and his colleague investigated a series of linear algorithms for ERP

detection and proposed hierarchical discriminant component analysis based on 50ms sliding non-overlapping window (Parra *et al*., 2005; Parra *et al*., 2008). In this chapter, we will exploit the positive aspects of both our baseline system and Parra et al's work and introduce a hierarchical window-classifier scheme followed by naïve Bayesian fusion.

## 8.2 Method

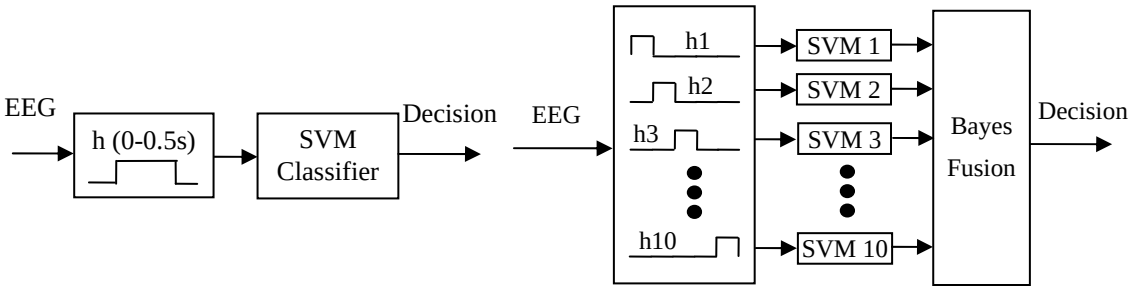



Figure 8-1 Block diagram of different windowing schemes for single trial ERP detection

In our baseline system, we extract features using EEG signals within 0~500ms window from the stimuli onset. With the 256 Hz sampling rate, the dimension of features is 129X32=4128. In our experiment, only one target is contained within each trial. Usually, each session contains 100-200 trials, i.e. 100-200 targets. This number is very low compared with the dimension. Moreover, ERPs usually appear between 150ms to 450ms from the stimuli onset, using all data within 0~500ms window may introduce irrelevant noise. This motivates us to consider different windowing scheme for feature extraction. The idea of a hierarchical window-classifier scheme is to do classification on multiple non-overlapping windows, and then fuse the classification result using Bayesian method. Figure 8-1 (b) shows a block diagram of the hierarchical window-classifier scheme. The windowing scheme used in our baseline system is shown in Figure 8-1 (a):

### 8.2.1 Feature Extraction

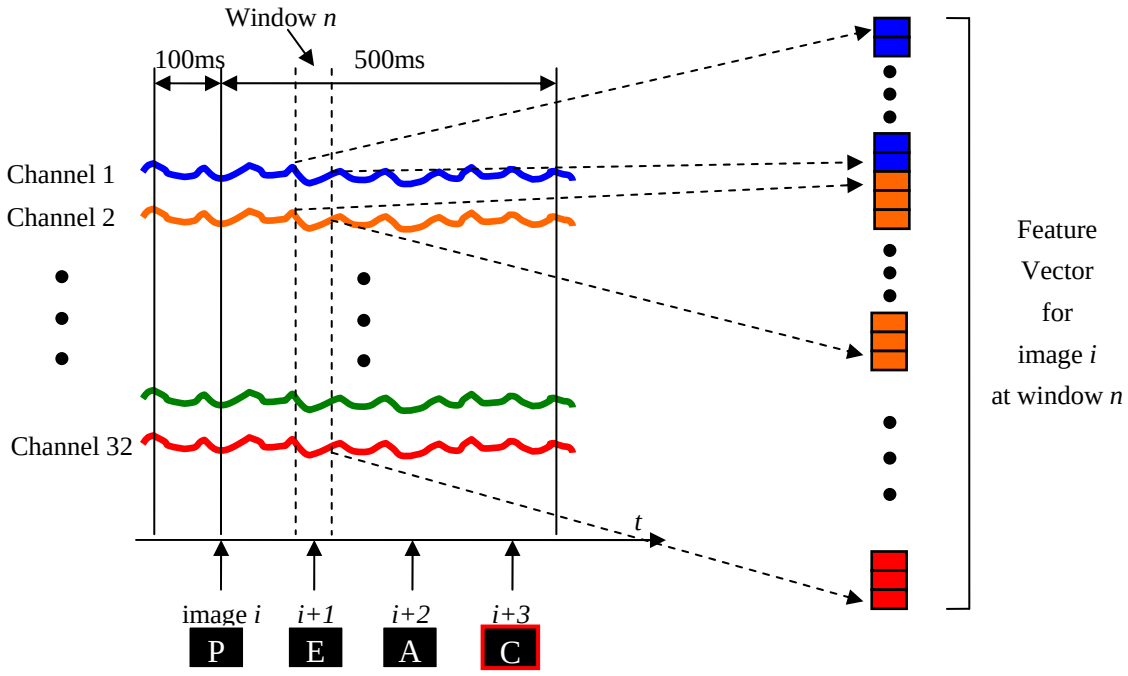



Figure 8-2 An example of feature extraction for window *n*

In section 7.2.2, we extracted features using 0~500ms window after the stimuli onset (Figure 7-3). The pre-stimulus window (-100ms~0) is used for normalization. However, in the hierarchical window scheme, an *l*-length non-overlapping sliding window is used to separate the 500ms window into multiple groups. Before we extract features, we basically follow the same procedure in our baseline system: truncate the signal using a 0~500ms window, then normalize the signal of each channel using the signals from pre-stimulus window (-100ms~0). After normalization, a sliding window is applied to the normalized data. We concatenate data from all channels within this sliding window, and get a feature vector within the sliding window. An example of feature extraction in window *n* is shown in Figure 8-2. After feature extraction,

features within each window are sent to a group of SVM classifiers, and the classification results are fused to get the final decision. We introduce the hierarchical SVM-Bayes ERP detector in the next section.

### 8.2.2 Hierarchical SVM-Bayes ERP Detector

As an alternative to classifying the vectorized raw data using a Gaussian-SVM, ten 50ms non-overlapping windows are used to partition the feature vector (in time, the interval 0,500[ms]) that are then individually classifier by Gaussian-SVMs of their own. The decisions of these first-layer SVMs are fused using the naïve-Bayes classifier approach, assuming that each binary decision is conditionally independent (given true label). Figure 8-1 (b) illustrates the structure of this hierarchical SVM-Bayes ERP detector. In (b), all ten windows are used. However, based on the prior knowledge, we know that ERP will appear in windows 5-9. So other than using all windows, we can also use a subset of windows. In this chapter, we use pre-defined window 5-9. However, the selection of window can also be done adaptively.

Assume we have $m$ windows, let the first layer SVM decisions be denoted by $d_j$ ($j$=1,…,$m$) and let $c$ be the true ERP label:

$$p_{j,kl} = p(d_j = k \mid c = l), \quad k,l = 0,1 \tag{Eq. 8-1}$$

Employing Bayes' rule and invoking the conditional independence assumption for the decision:

$$p(c \mid d_1 ... d_m) \propto p(d_1 ... d_m \mid c)\, p(c) = p(c) \prod_{j=1}^{m} p(d_j \mid c) \tag{Eq. 8-2}$$

For equal risks, defining the discriminant threshold as $th$=$p$($c$=0)/$p$($c$=1), according to Bayes rule, the naïve Bayes decision-level fusion rule becomes:

$$\prod_{j=1}^{m} p(d_j \mid c = 1) \Big/ \prod_{j=1}^{m} p(d_j \mid c = 0) \overset{c=1}{>} th \tag{Eq. 8-3}$$

Combining (1) and (3) and noting that the threshold can be modified for different risk-ratios for miss and false detections:

$$\prod_{j=1}^{m} p_{j01}^{(1-d_j)} p_{j11}^{d_j} \Big/ \prod_{j=1}^{m} p_{j00}^{(1-d_j)} p_{j10}^{d_j} \overset{c=1}{>} th \qquad \text{(Eq. 8-4)}$$

where we estimate $p_{j00}$, $p_{j01}$, $p_{j10}$, $p_{j11}$ from the training set (via cross validation), and obtain $d_j$ from layer-one SVM classifiers for each testing sample.

The fact that P300 happens around 300ms after stimulus onset implies that not all windows carry useful information. For example, window 1 ([0,50]ms) carries much stronger background activity than any potentially present stimulus-related response or even transients from the stimulus switching boundaries, which is expected to compromise the ERP detection accuracy. Eliminating irrelevant windows (features) is expected to improve performance and make the detector robust. Based on these observations, windows 5-9 ([200,450]ms) are expected to carry most discriminative energy. Two variations of the SVM-Bayes detector described above and compared in our study use (i) all 10 windows, and (ii) only windows 5-9 during fusion.

## 8.3 Experimental Results

We applied the three ERP detector schemes described above to the EEG data collected from four subjects. Each subject finished eight sessions of experiments in four days, with two sessions per day. We used the data from the morning sessions as the training set and the data from afternoon sessions on the same day as the testing set. The results are represented using ROC curves, the area under ROC curves, and the MFAR (minimum false alarm rate at zero miss).

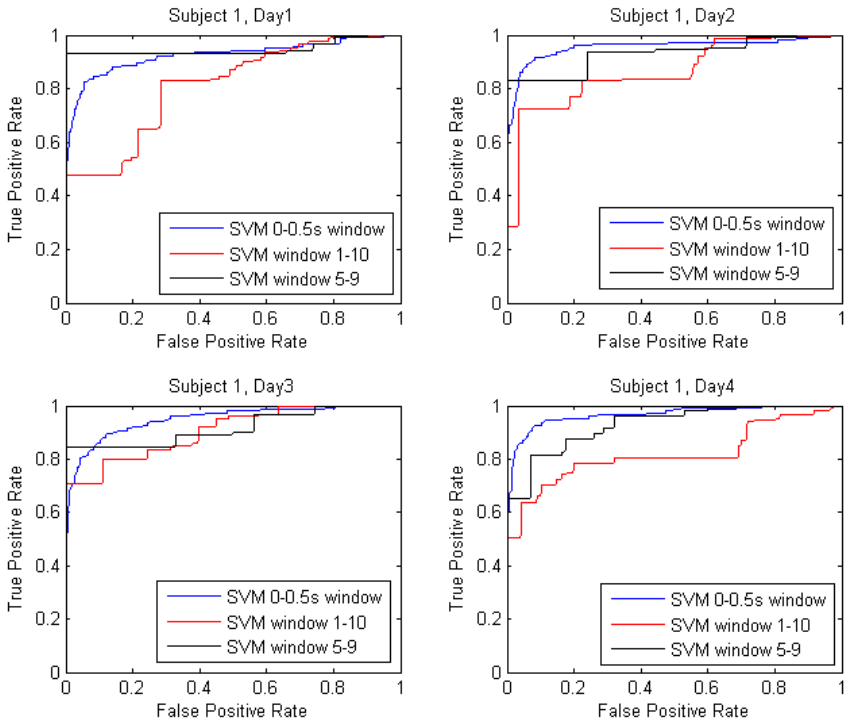



Figure 8-3 ROC curves for four days data using three different ERP detection schemes from subject 1.

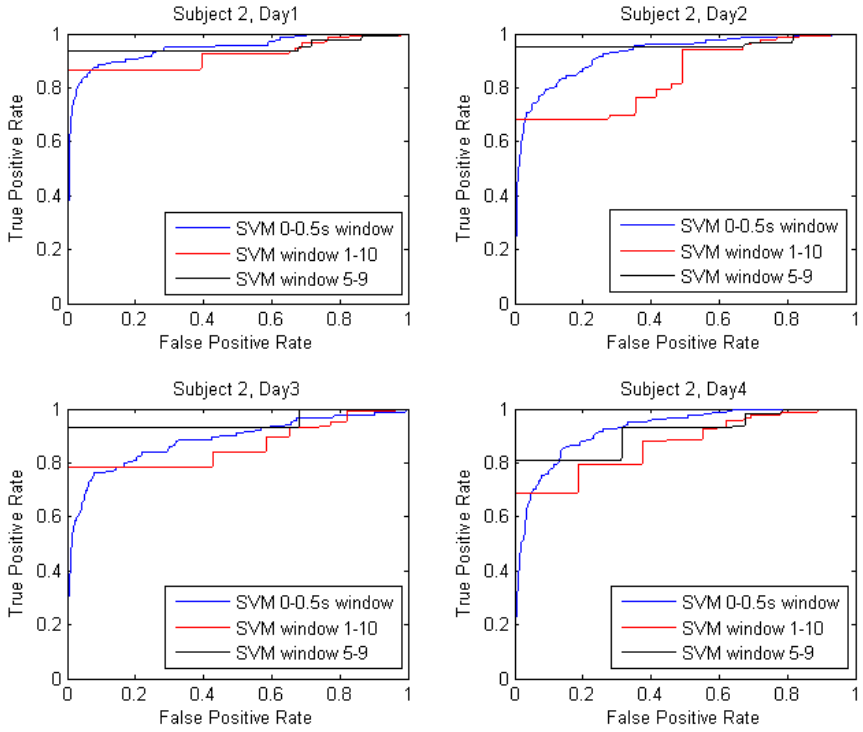



Figure 8-4 ROC curves from four days data using three different ERP detection schemes for subject 2.

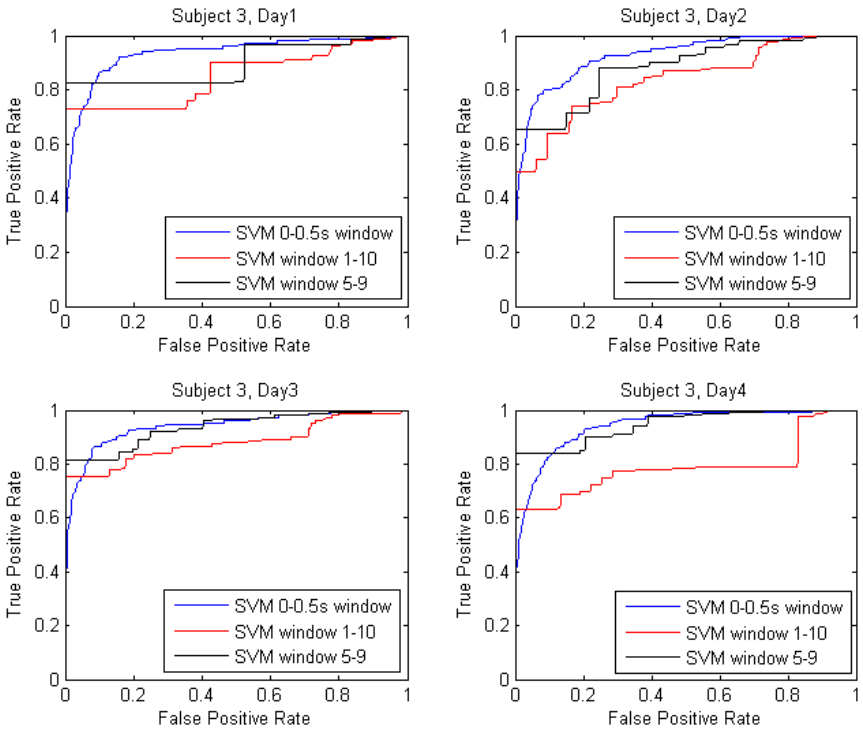



Figure 8-5 ROC curves from four days data using three different ERP detection schemes for subject 3.

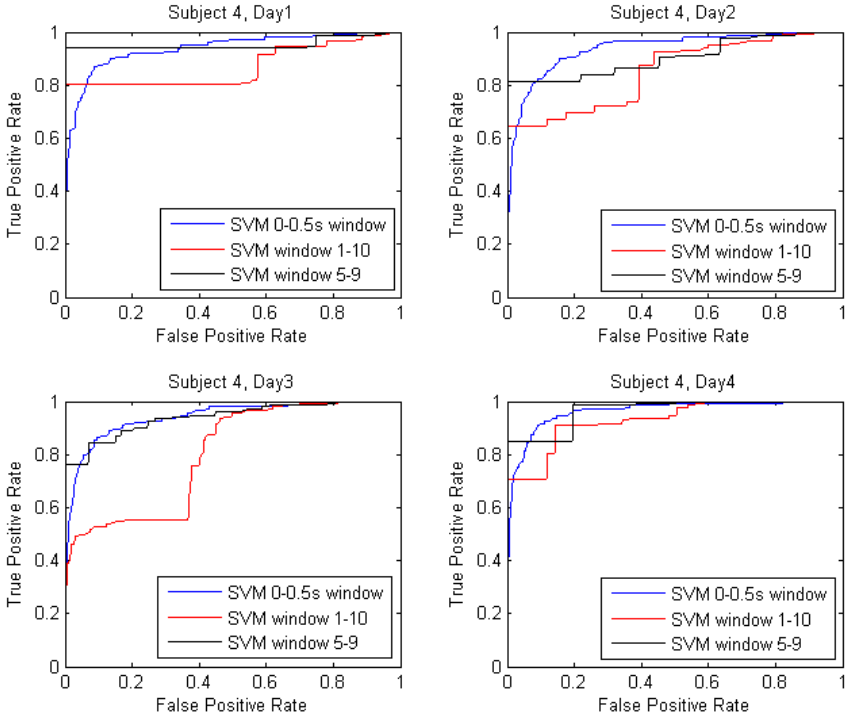



Figure 8-6 ROC curves from four days data using three different ERP detection schemes for subject 4.

Table 8-1 Area under curves for four days data using three different ERP detection schemes from subject 1

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| 0-0.5s window | 0.92 | 0.96 | 0.95 | 0.97 |
| window 1-10 | 0.81 | 0.86 | 0.91 | 0.82 |
| window 5-9 | 0.95 | 0.93 | 0.92 | 0.93 |

Table 8-2 Area under curves for four days data using three different ERP detection schemes from subject 2.

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| 0-0.5s window | 0.95 | 0.93 | 0.88 | 0.92 |
| window 1-10 | 0.92 | 0.84 | 0.86 | 0.87 |
| window 5-9 | 0.95 | 0.96 | 0.95 | 0.91 |

Table 8-3 Area under curves for four days data using three different ERP detection schemes from subject 3.

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| 0-0.5s window | 0.93 | 0.93 | 0.94 | 0.94 |
| window 1-10 | 0.85 | 0.84 | 0.88 | 0.79 |
| window 5-9 | 0.90 | 0.88 | 0.94 | 0.95 |

Table 8-4 Area under curves for four days data using three different ERP detection schemes from subject 4.

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| 0-0.5s window | 0.94 | 0.94 | 0.94 | 0.96 |
| window 1-10 | 0.87 | 0.85 | 0.80 | 0.93 |
| window 5-9 | 0.95 | 0.91 | 0.94 | 0.97 |

The results are comprehensively presented in figures and tables. Figure 8-3 – Figure 8-6 show ROC curves for each of the four subjects in four days. Table 8-1 – Table 8-4 show the area under ROC curves for each subject in four days, and Tables 8-5 – Table 8-8 show the MFAR for each subject in four. While we do not observe that one method is clearly superior to the other two in all the instances, the following qualitative observations emerge: (i) In general, using windows 5-9 in the SVM classification, followed by naïve Bayes fusion of their decisions, performs better more often than using an SVM on the whole [0,500]ms window, and these two are better almost all the time from than fusing SVM decisions of all 10 windows; (ii) the naïve fusion of decisions on windows 5-9 approaches zero-false alarm rate at a higher

detection rate than SVM on [0,500]ms, while the latter approaches 100% detection rate (0-miss) at a lower MFAR, based on the ROC curves on finite amount of trials used to obtain these results.

Table 8-5 MFAR for four days data using three different ERP detection schemes from subject 1

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| 0-0.5s window | 0.95 | 0.96 | 0.81 | 0.77 |
| window 1-10 | 0.91 | 0.88 | 0.68 | 0.98 |
| window 5-9 | 0.95 | 0.99 | 0.87 | 0.81 |

Table 8-6 MFAR for four days data using three different ERP detection schemes from subject 2.

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| 0-0.5s window | 0.71 | 0.93 | 1.00 | 0.64 |
| window 1-10 | 0.99 | 0.93 | 0.97 | 0.89 |
| window 5-9 | 1.00 | 0.95 | 0.94 | 0.82 |

Table 8-7 MFAR for four days data using three different ERP detection schemes from subject 3.

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| 0-0.5s window | 0.96 | 0.66 | 0.85 | 0.87 |
| window 1-10 | 0.97 | 0.85 | 0.99 | 0.93 |
| window 5-9 | 0.99 | 0.89 | 0.91 | 0.75 |

Table 8-8 MFAR for four days data using three different ERP detection schemes from subject 4.

| | Day 1 | Day 2 | Day 3 | Day 4 |
|---|---|---|---|---|
| 0-0.5s window | 0.87 | 0.83 | 0.81 | 0.82 |
| window 1-10 | 0.97 | 0.93 | 0.82 | 0.62 |
| window 5-9 | 0.98 | 0.90 | 0.82 | 0.69 |

## 8.4 Discussion

In this chapter, we described three schemes for single ERP detection. Using the Gaussian-SVM on temporal window [0,500]ms as a baseline, we evaluated two decision level Bayesian fusion approaches that utilize short-window SVM decisions of the ERP waveform following stimulus onset (inspired by the

success reported by Parra and Sajda). Experimental results on four subjects across sessions illustrate that SVM on [0,500]ms (scheme 1) and SVMs on windows at [200-450]ms fused at decision level via naïve Bayes method (scheme 3) yield similar performances, while fusing the decisions of all ten windows in the [0,500]ms post-stimulus interval (scheme 2) is inferior. This is expected since scheme 2 corrupts the accuracy by introducing decisions from temporal data that do not contain sufficiently powerful evidence about the presence or lack of ERP waveforms.

Although scheme 3 does not exhibit better performance than scheme 1 in all instances we analyzed, it has several advantages. First, it utilizes a smaller dimensionality feature vector, and thus is expected to be more robust over long-term BCI training; this also contributes to computational efficiency for real-time implementation. Second, by breaking the raw data into multiple discriminant temporal components, the EEG channel selection becomes feasible for each short window; this is important because at different phases of the response, different regions of the brain become active, therefore for each window, appropriate channels can be retained, contributing to further reduction of feature dimensionality, hence classifier robustness. We can use the mutual information based EEG channel selection method described in Chapter 4 for each short temporal window in order to achieve this goal. Third, scheme 3 allows us to further improve the Bayesian fusion model utilizing more elaborate graphical models of dependencies between the temporal windows. Even with the naïve Bayesian fusion, this approach is competitive with our baseline approach in the experiments performed.

The temporal windowing scheme provides us the opportunity to select information in the time domain. In the next chapter, we focus on identifying informative features in the frequency domain and the spatial domain.

# Chapter 9: Identifying Informative Features in the Frequency Domain and the Spatial Domain for Single Trial ERP Detection

In Chapter 8, we introduced different temporal windowing schemes for single trial ERP detector in order to identify informative features in the time domain. Although the described method is not obviously superior to our baseline system, it has some advantages: 1) it reduces the dimension, hence more robust and computational efficient for real-time implementation; and 2) it allows us to do further feature selection and dimensionality reduction using the existing mutual information based approach which we described in Chapters 4 and 5. In this chapter, we will expand our study to the frequency domain and the spatial domain. Different feature extraction and selection strategy will be discussed in this Chapter. This chapter is based on publication (Lan *et al*., 2010a).

## 9.1 Background

Our previous study on a single-trial ERP detector showed that a Support Vector Machine (SVM) based classifier yielded better performance than other classifiers (Huang *et al*., 2008; Lan *et al*., 2009). However, Krusienski compared different classifiers and found that the SVM was not necessarily the best classifier in a traditional P300 speller systems (Krusienski *et al*., 2006). Krusienski also found that given a properly selected EEG channel subset, an LDA based classifier outperformed the SVM classifier in the target detection accuracy and robustness. Moreover, compared to SVM classifiers, LDA classifiers require far less computation in training, making them more suitable for adaptive real-time systems. In Krusienski's study, pre-determined channel subsets were compared and used throughout the study. Though Krusienski

claimed the selected subset exhibited session-to-session transfer ability, no explicit results demonstrated that this selection also transferred between subjects. An online feature selection approach is more desirable for real world applications. In this chapter, we focused our study on feature selection in both the frequency domain (frequency bands) and the spatial domain (EEG channels). We presented English letters in an RSVP paradigm, collected EEG data, and analyzed them offline using an LDA classifier.

## 9.2 Identify Informative Features in the Frequency Domain

EEG data was collected using the same method we described in Chapter 7. After collecting EEG signals, we first filtered EEG signals using a bandpass filter. Previously, we used a 1-45Hz passband filter (Huang *et al*., 2008; Lan *et al*., 2009). In the current study, we compared the ERP energy in narrow bands ranging from 0 to 44Hz, and determined that the ERP energy was mainly concentrated in the 0-1Hz and 10-20Hz bands. Since the P300 has limited energy in the 10-20Hz band, this indicates that it is not the sole contributor to the ERP based detection. Based on the same analyses, we determined that the 0-20Hz (no DC) passband yielded the best performance. Consequently, this filter was used throughout.

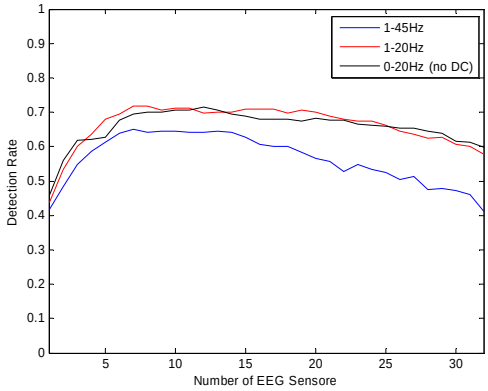
The filtered data were truncated using a [0, 500ms] window following each image stimulus (called an "epoch" in what follows) and normalized with the [-100ms, 0] pre-stimulus window. We concatenated data within one epoch by channels, and to obtain a data point with 32×129=4128 dimensions (each channel contained 129 samples).

## 9.3 Identify Informative Features in the Spatial Domain

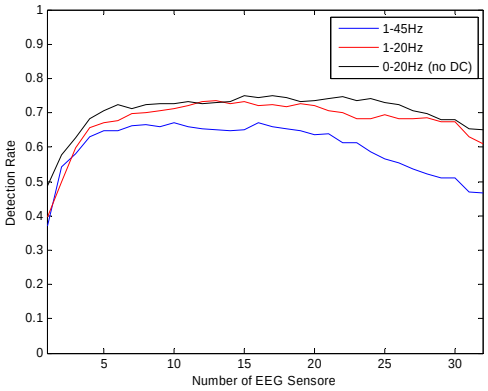
EEG channels were ranked using the wrapper approach (error based approach) with a greedy search strategy. For a given channel subset, all samples from these channels were concatenated to form a new data point for each epoch. We used epochs from the first 50 trials as the training set, used the remaining data as the testing set, applied the LDA classifier on three sequences separately, and fused results using a majority vote for the final decision. The accuracy of this final decision was used as criterion for ranking the EEG channels. The channel ranking results are shown in Figure 9-1 (a) – (f). Each figure corresponds to a subject/session. Three curves of different colors denote features in three different frequency bands. As mentioned, a frequency range of 0-20Hz with no DC (black curve) yielded the best performance. The horizontal axis denotes the number of optimal channels used for ERP detection; the vertical axis denotes the detection accuracy.

Theoretically, the detection accuracy should increase as more EEG channels are used since more information is available. However, due to noise and the finite amount of training data, using more channels beyond a critical number causes over-fitting and hence reduces performance. This explains why in Figure 9-1 the optimal performance is obtained when less than 20 channels are selected.

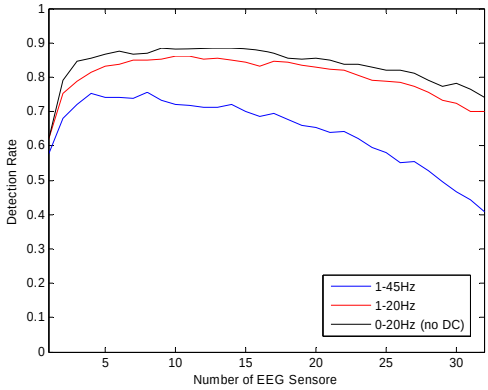
We note that determining an optimal set in the original high dimensional feature space is time-consuming and can only be done offline. If session-to-session transfer and subject-to-subject transfer were unproblematic, one could find the optimal subset of channels offline and then use them in the online system. However, transfer appears to be an issue. For example, the optimal channel subset for subject 1, session 1 was: F7, O1, Fp2, P7, Pz, PO3, AF4, CP2, P3, T8, FC1, FC2, while the optimal channel subset for subject 2, session 1 was: PO3, T7, Oz, CP2, Cz, F8, AF3, O2, P8, C4, P7, F4. This points out the need for fast feature selection algorithms in real-time BCIs.



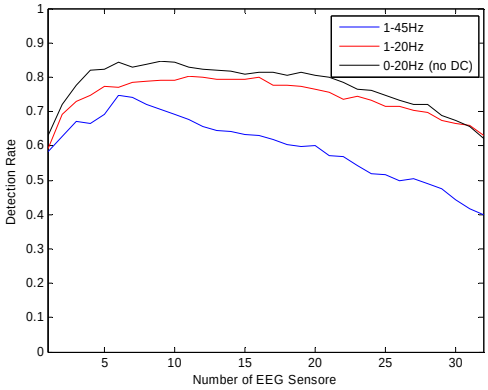

(a) Subject 1, Session 1.



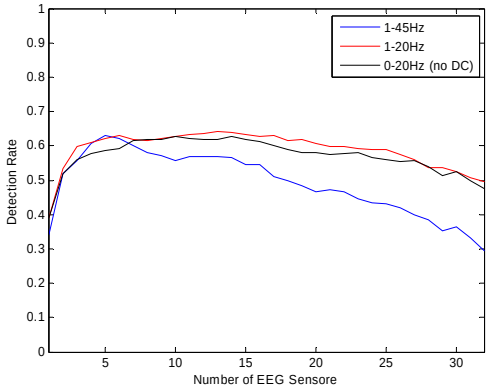

(b) Subject 1, Session 2



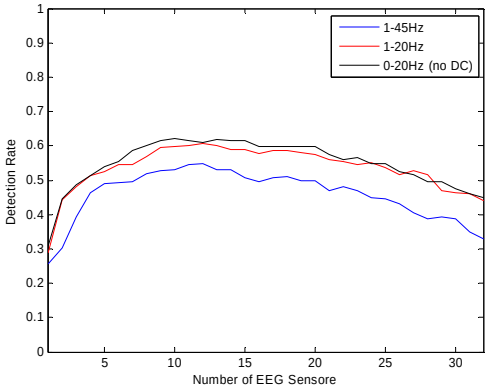

(c) Subject 2, Session 1.




(d) Subject 2, Session 2




(e) Subject 3, Session 1.



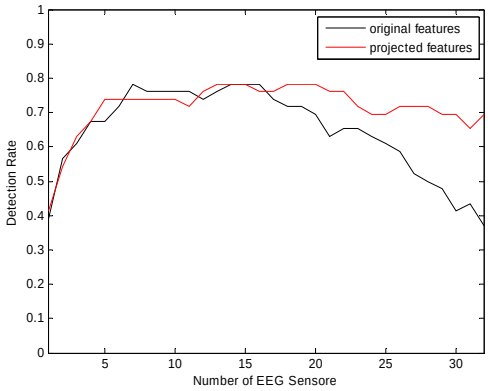

(f) Subject 3, Session 2

Figure 9-1 Channel ranking results for three subjects, in two sessions with different frequency passbands

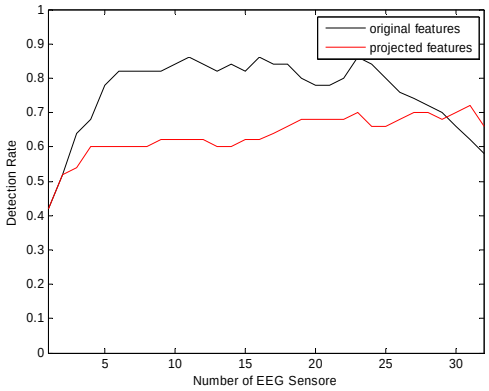
## 9.4 Channel-wise Dimensionality Reduction

It is not feasible to do online channel ranking and subset channel selection in high dimension due to the computational cost, making the dimension reduction critical. Subspace feature projection approaches are widely used for dimensionality reduction, such as PCA, ICA, and LDA. We compared different subspace projection methods and found that LDA yielded the best performance, as expected. Thus, prior to channel ranking, we applied dimensionality reduction on each channel separately using LDA, which reduced the dimension from 4128 to 32, with one feature per channel. As a result, it only seconds to rank channels in the projected feature space.

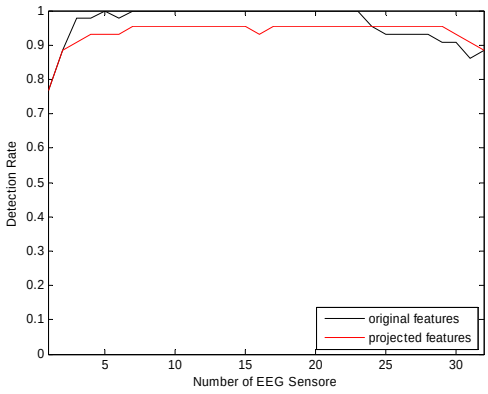
The performances of channel ranking with original features and projected features are compared and shown in Figure 9-2 (a) – (f). Each figure contains two curves with different colors, denoting different channel ranking methods. As expected, the overall performances of using the original features were better than using the projected features, since projecting from 129 dimensions to 1 dimension on each channel also eliminated useful information. However, we observe that the best performances of a subset channels ranked from the projected features were generally better than using all channels on the original features. This indicates that for online real-time BCI systems, dimensionality reduction combined with channel ranking can present a better tradeoff when the online channel selection using the original feature space is not feasible.



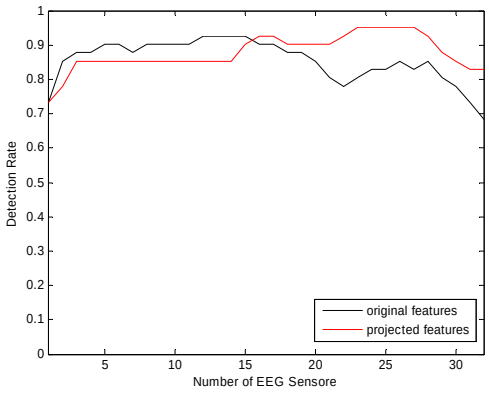

(a) Subject 1, Session 1.



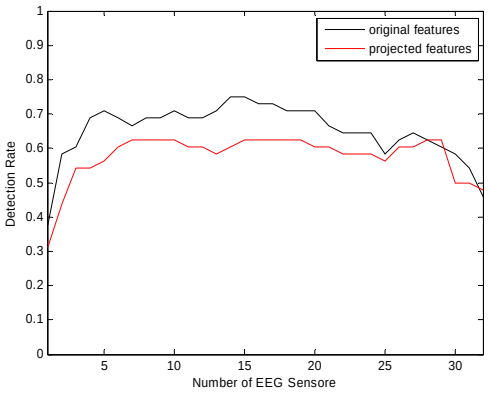

(b) Subject 1, Session 2



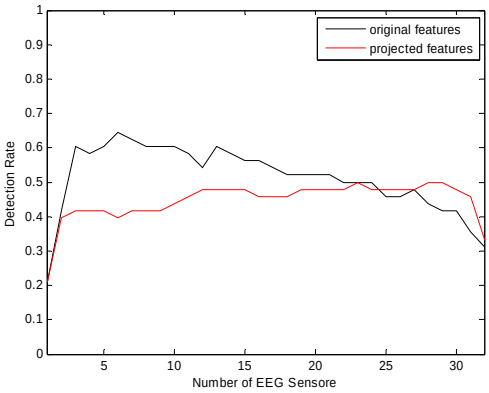

(c) Subject 2, Session 1.




(d) Subject 2, Session 2




(e) Subject 3, Session 1.




(f) Subject 3, Session 2

Figure 9-2 Channel ranking results using original features and LDA projected features on three subjects, in two sessions.

## 9.5 Summary

In this chapter, we explored the frequency domain and the spatial domain of EEG signals, and compared different features extracted and selected from a range of frequency bands and EEG channel subsets.

Experimental results indicated that the ERP energy is concentrated in the 0-1Hz and 10-20Hz bands. This suggests that not only P300, but also other ERPs are relevant for ERP based communication. The results were consistent across subjects and sessions, although we did not have enough subjects to employ statistical confidence tests.

The performance of the error based feature ranking approach indicated that not all EEG channels were needed for ERP detection. Using a subset of EEG channels yielded better performance than using all channels. Furthermore, this subset was different across subjects. Ranking channels using the original features gave accurate results, but it was computation-intensive and not suitable for real-time system. Channel-wise dimensionality reduction followed by ranking channels using the projected features dramatically reduces the computational needs. Although this approach sacrifices accuracy, the results were still better when channel selection was performed. We may improve the approach by employing other subspace projection methods instead of using LDA. In the next chapter, we compare several well-defined subspace projection methods to LDA projection we used in this chapter, and find the most suitable projection that yields the best ERP detection accuracy.

# Chapter 10: Compare Different Subspace Projection Methods for Feature Selection in Single Trial ERP Detection System

In Chapter 9, we explored the frequency and spatial domain of EEG signals, and selected frequency bands that contains most of the ERP energy. We also used an error-based wrapper approach to select EEG channels. However, since the dimension of the feature vectors is high (129X32=4128), even with a greedy search strategy and LDA classifiers, the channel selection procedure is computational intensive. We also introduced the concept of Channel-wise dimensionality reduction. Instead of ranking EEG channels in raw feature format directly, we first applied LDA projection to all features of each channel to reduce the dimension; and then ranked EEG channels in the low dimensional space. This greatly reduced the computational load of the system. However, the channel-wise LDA projection reduced the ERP detection accuracy as well. In this chapter, we explore other subspace projection methods and compare them with the performance of LDA projection. This chapter is based on publication (Lan *et al*., 2010b).

## 10.1 Background

In Chapter 9, we introduced the wrapper approach for EEG channel selection coupled with an LDA classifier, and applied this method on the EEG data from different subjects. To accelerate the channel ranking procedure, we also proposed a method which projected features of each channel into low dimensional space first before ranking them. The block diagram of this channel-wise dimensionality reduction and EEG channel selection method is shown in Figure 10-1.

We use the same procedure for preprocessing and feature extraction as described in Section 9.2. After getting raw features, we applied different subspace projection methods on the features of each channel first, which greatly reduces the dimension. In Chapter 9, we used LDA projection for this purpose. In this chapter, we explorer other subspace projection methods: 1) Principal Component Analysis (PCA), 2) Sparse PCA (SPCA), 3) Empirical Mode Decomposition (EMD), and 4) Local Mean Decomposition (LMD). After channel-wise dimensionality reduction, we rank the EEG channels in the projected low-dimensionality feature space. In the next section, we will briefly describe the selected subspace projection methods. And experimental results will be shown in the section after that.

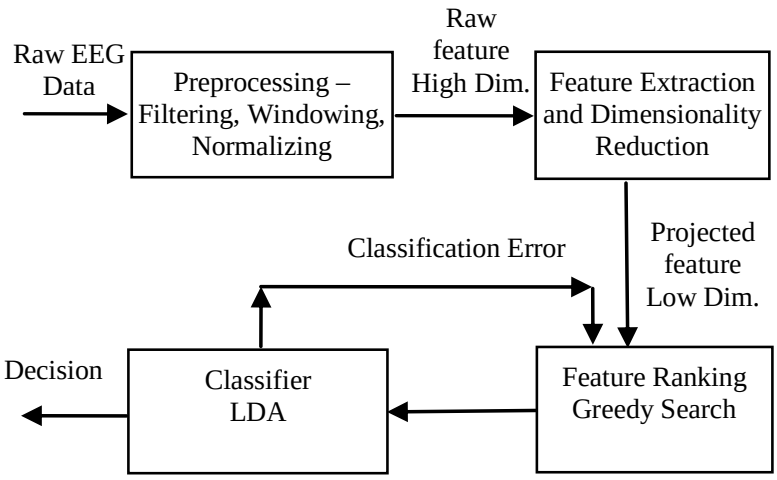



Figure 10-1 Block diagram of channel-wise dimensionality reduction and EEG channel selection

## 10.2 Different Subspace Projection Methods

There are several well-developed subspace projection methods. In this chapter, other than LDA, we are going to investigate other subspace projection methods for channel-wise dimensionality reduction, including Principal Component Analysis (PCA), Sparse PCA (SPCA), Empirical Mode Decomposition (EMD), and Local Mean Decomposition (LMD), and compare results with those of LDA in Chapter 9.

### 10.2.1 Principal Component Analysis (PCA)

PCA is probably one of the most commonly used dimensionality reduction methods. PCA finds a linear combination of the multivariate data that captures a maximum amount of variance. However, the projections that PCA seeks are not necessarily related to the class labels; hence they may not be optimal for classification problems. In this study, we apply PCA to the features of individual EEG channels, and then compare the channel ranking performance using the first 1, 5 and 10 components of each channel (in our particular application, typically, using 5 components carries 75% variance, and using 10 components carries 90% variance).

### 10.2.2 Sparse Principal Component Analysis (SPCA)

PCA has many advantages, such as that it captures maximum variance, and the components are uncorrelated. However, the principal components are usually linear combinations of all variables. Hence it is not possible to discover a low dimensional space that explains most of the variance. For example, we apply PCA on individual EEG channels for a 0.5s window of data (129 points) after the stimuli onset. This projection can not determine which points (when after the stimuli onset) contain more information for ERP detection. It would be interesting to discover sparse principal components by sacrificing some of the explained variance and the orthogonality. Among many existing SPCA algorithms, we choose DSPCA in our study (d'Aspremont *et al*., 2007), and compare the channel ranking performance using the first 1, 5 and 10 components of each channel.

### 10.2.3 Empirical Mode Decomposition (EMD)

EMD was first proposed by Huang et al. (1998) for analyzing signals of nonlinear and nonstationary time series. EMD can be used to decompose any time series into a finite number of functions called intrinsic mode functions (IMFs) without leaving the time domain. The Intrinsic Mode Functions are nearly orthogonal and sufficient to describe the signal. Unlike other time-frequency methods, such as short time Fourier transform and wavelet transform, EMD does not make any assumptions about the data. Hence it is more flexible in extracting time-frequency information from EEG data.

EMD finds a local mean envelope by creating maximum and minimum envelopes around the signal using cubic spline interpolation through the individual local extrema. The mean envelope, the half sum of the upper and lower envelopes, is then subtracted from the original signal, and the same interpolation scheme is iterated on the remaining signal. This is called the "sifting" process (SP). SP terminates when the mean envelope is approximately zero everywhere, and the resultant signal is designated as an IMF. After the first IMF is removed from the original data, the next IMF is extracted iteratively by applying the same procedure (Table 10-1).

As given by the nature of this decomposition procedure, the data are decomposed into *n* fundamental components, each with a distinct time scale. More specifically, the first component associates with the smallest time scale, which corresponds to the fastest time variation of data. As the decomposition process proceeds, the time scale increases, and hence, the mean frequency of the mode decreases. By combining different IMFs, EMD can be used as low-pass, high-pass, or band-pass filters.

Although after applying EMD, different feature extraction methods can be further applied, in this study, we remove the high frequency component of IMFs (based on the assumption that ERP energy concentrates in lower frequencies). Since the IMFs have the same length as original data, we apply PCA on

IMFs to reduce the dimension. This procedure is repeated for all EEG channels, and the channel ranking performance is evaluated using an LDA classifier.

Table 10-1 EMD algorithm for channel-wise dimensionality reduction

- The EMD will break down a signal into its component IMFs.
- An IMF is a function that:
  1. has only one extreme between zero crossings, and
  2. has a mean value of zero.
- The IMFs is acquired by sifting process:
  1. For a signal $X(t)$, let $m_1$ be the mean of its upper and lower envelopes as determined from a cubic-spline interpolation of local maxima and minima.
  2. The first component $h_1$ is computed: $h_1=X(t)-m_1$
  3. In the second sifting process, $h1$ is treated as the data, and $m_{11}$ is the mean of $h_1$'s upper and lower envelopes: $h_{11}=h_1-m_{11}$
  4. This sifting procedure is repeated $k$ times, until $h_{1k}$ is an IMF, that is: $h_{1(k-1)}-m_{1k}=h_{1k}$
  5. Then it is designated as $c_1=h_{1k}$, the first IMF component from the data, which contains the shortest period component of the signal. We separate it from the rest of the data: $X(t)-c_1=r_1$ The procedure is repeated on $r_j$: $r_1-c_2=r_2,....,r_{n-1}-c_n=r_n$.

### 10.2.4 Local Mean Decomposition (LMD)

The local mean decomposition (LMD) was developed recently by Smith (Smith 2005) to decompose amplitude and frequency modulated signals into a small set of product functions, each of which is the product of an envelope signal and a frequency modulated signal from which a time-varying instantaneous phase and instantaneous frequency can be derived. Similar to EMD, LMD decomposes data into a series of functions in the time domain. It does not require any assumption about the data. Unlike that EMD uses

cubic splines, which may induce information loss, LMD uses the smoothed local means (moving average filter) to determine a more credible and reliable instantaneous frequency directly from the oscillations within the signal. The LMD algorithm is shown in Appendix B. In our study, we apply LMD and evaluate the performance using the same method as we explained above in EMD section.

Table 10-2 LMD algorithm for channel-wise dimensionality reduction

1. From the original signal $x(t)$, determine the mean value, $m_{i,k,}$ and local magnitude, $a_{i,k}$, with extrema, $n_{k,c}$ ($t$: time, $i$ : number of Product Function, $k$ : iteration number in a process of Product Function, $c$ : sequence of the extrema)

   $$m_{i,k,c}= (n_{k,c} + n_{k,\,c-1})/2, \quad a_{i,k} = | n_k - n_{k+1}|/2$$

2. Interpolate straight lines of mean (local magnitude) values between successive extrema.
3. Smooth the interpolated signal using moving average filter.
4. Subtract the smoothed mean signal from the original signal, $x(t)$.

   $$h_{i,k}(t) = x(t) - m_{i,k}(t)$$

5. Get the frequency modulated signal, $s_{i,k}(t)$, by dividing $h_{i,k}(t)$ by $a_{i,k}(t)$.

   $$s_{i,k}(t) = h_{i,k}(t)/a_{i,k}(t)$$

6. Check whether the $a_{i,k}(t)$ is equal to 1 or not.
7. If not, multiply $a_{i,k}(t)$ by $a_{i,k-1}(t)$and go to the first step.
8. Envelope, $a_i(t)$, can be derived by multiplying the whole $a_{i,k}(t)$ until $a_{i,k}(t)$equals one.

   $$a_i(t) = a_{i,1}(t) \times a_{i,2}(t) \times a_{i,3}(t) \times \ldots \times a_{i,l}(t)$$

   (l : maximum iteration number)
9. Derive Product Function by multiplying $a_i(t)$ by $s_{i,l}(t)$

   $$PF_i = a_i(t) \times s_{i,l}(t)$$

10. Subtract $PF_i(t)$ from $x(t)$, and then go to the first step with the remainder.

## 10.3 Experimental Results

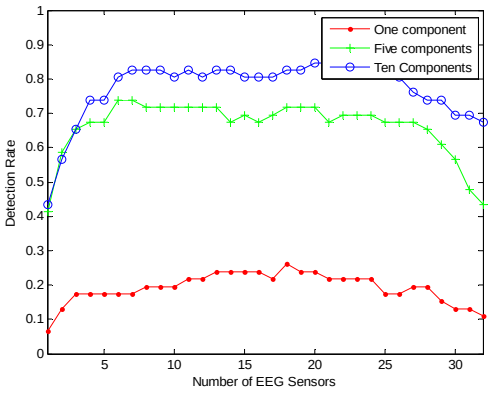
In PCA and SPCA projections, we want to find out the optimal number of projected features first.

### 10.3.1 PCA and SPCA with different number of components

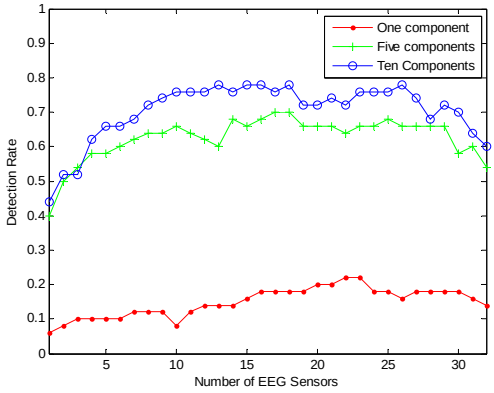
We apply both PCA and SPCA projections to individual EEG channels, and use the first 1, 5 and 10 components of each channel as features, then rank the EEG channels using the LDA classifier error. The feature ranking results for different subjects and sessions using different numbers of PCA components are compared, and figures are shown in Figure 10-2 (a) – (f). The similar results for SPCA are shown in Figure 10-3 (a) – (f). As we expected, using the first 10 components of each channel for both PCA and SPCA yields the best performances. However, the computation time using PCA is reasonably lower than using SPCA. It is interesting to see that when using only one component, SPCA performs better than PCA. However, the performances of PCA when using the first 5 and 10 components are better than that of SPCA. This indicates that using SPCA does not benefit us given the computational costs and performances combination. We will use 10 PCA components from each channel as features. For the rest of this study, we compare the results of other channel-wise dimensionality reduction with that of using 10 PCA components.

### 10.3.2 Compare Different Subspace Projection Methods

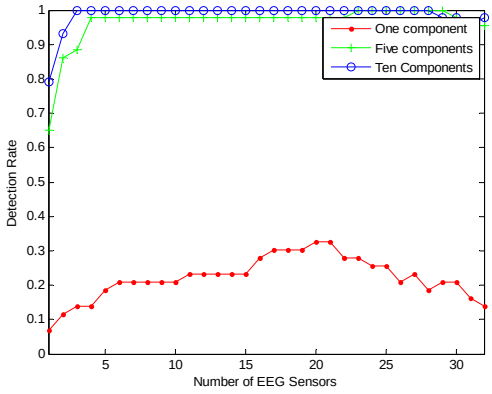
We compare feature ranking results using different dimensionality reduction methods coupled with error based channel selection methods: 1) using original features; 2) using channel-wise LDA projection for dimensionality reduction; 3) using channel-wise PCA projection, picking the first 10 components for each channel; 4) using channel-wise SPCA projection, picking the first 10 components for each channel; 5)



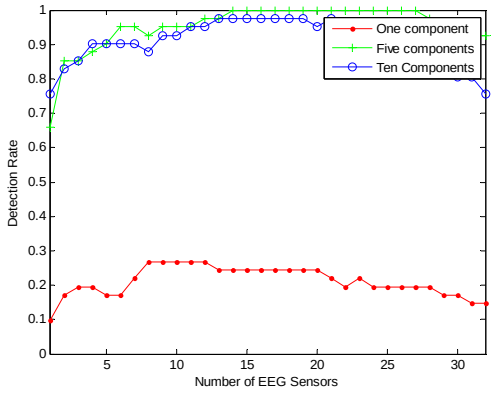

(a) Subject 1, Session 1.



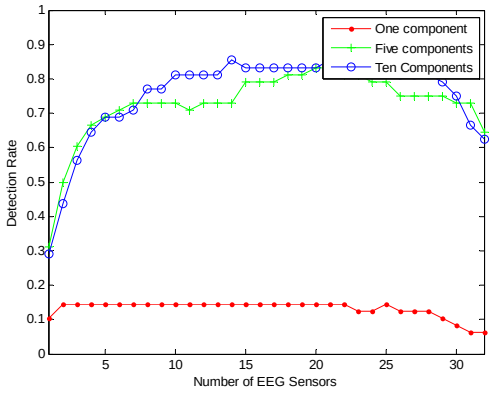

(b) Subject 1, Session 2



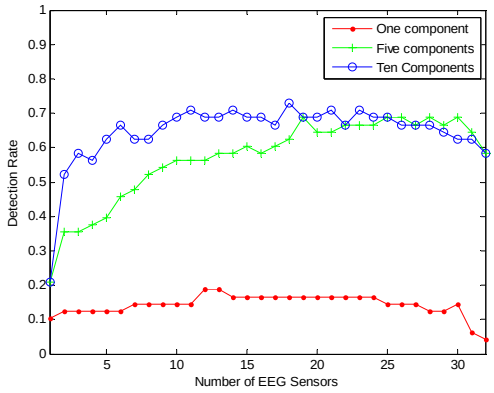

(c) Subject 2, Session 1.




(d) Subject 2, Session 2

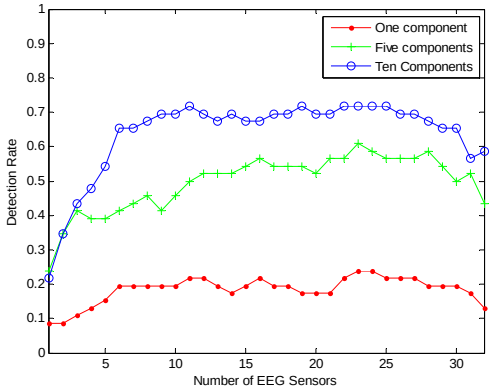



(e) Subject 3, Session 1.

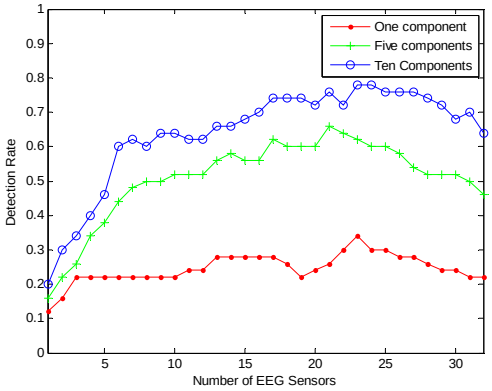



(f) Subject 3, Session 2

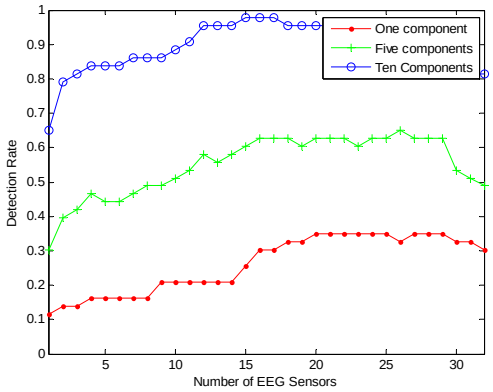
Figure 10-2 Feature ranking results for three subjects, in two sessions using different number of PCA components for the channel-wise dimensionality reduction and channel selection methods.



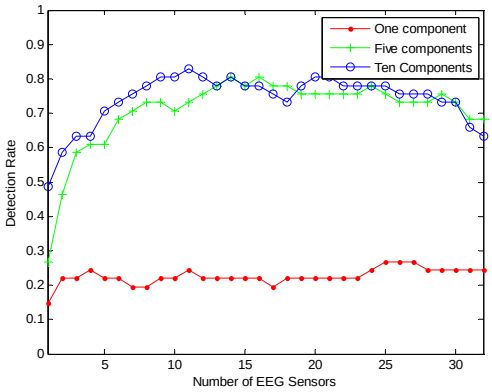

(a) Subject 1, Session 1.



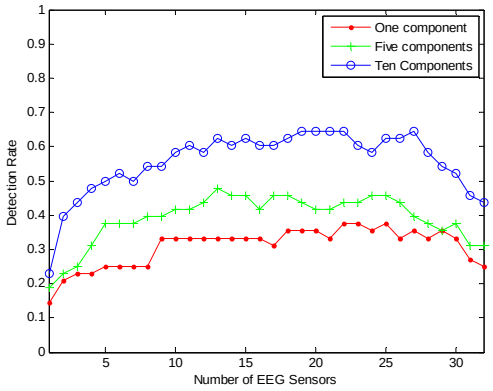

(b) Subject 1, Session 2



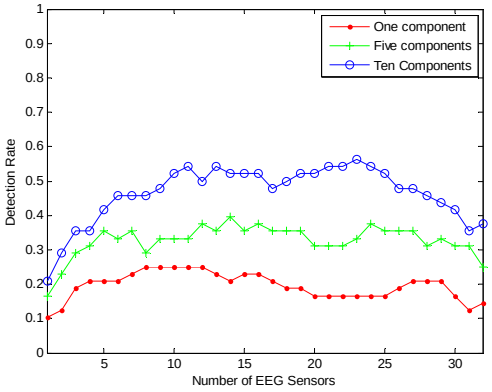

(c) Subject 2, Session 1.




(d) Subject 2, Session 2




(e) Subject 3, Session 1.




(f) Subject 3, Session 2

Figure 10-3 Feature ranking results for three subjects, in two sessions using different number of SPCA components for the channel-wise dimensionality reduction and channel selection methods.

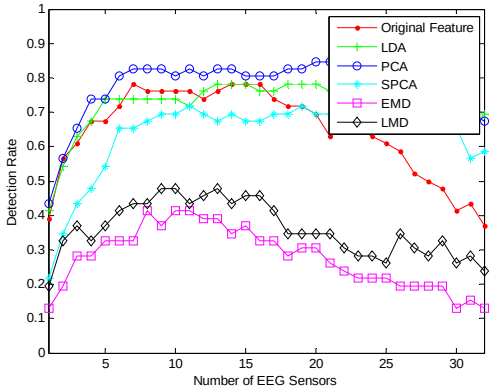
using channel-wise EMD for feature extraction and the first 10 PCA components of each channel for dimensionality reduction; and 6) using channel-wise LMD for feature extraction and the first 10 PCA components of each channel for dimensionality reduction. The results for different subjects/sessions using different methods are shown in Figure 10-4 (a) – (f). The performance of using original features and using the first 10 principal component of PCA are comparable. However, using PCA for dimensionality reduction is much faster than using original features. The performance of using channel-wise LDA for dimensionality reduction is acceptable, and it is the fastest method. The performances of using SPCA, EMD and LMD are the worst, and they generally cost more computational time.

## 10.4 Discussion

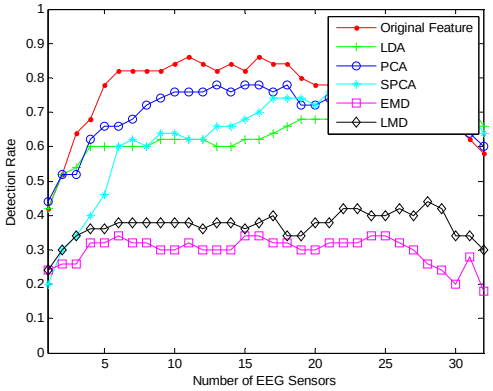
In this chapter, we compared different subspace projection methods for channel-wise dimensionality reduction and feature selection. Experimental results show that using original features and using PCA with the first 10 principal components for each channel perform the best, while the time-consuming SPCA, EMD, and LMD methods perform surprisingly worse, even for an offline system. The performance of LDA projection is acceptable and is the fastest method. Thus, we conclude that:

1) Channel-wise PCA projection with the first 10 principal components of each channel offers the best trade off in terms of accuracy and speed. It can be used in both online and offline systems.

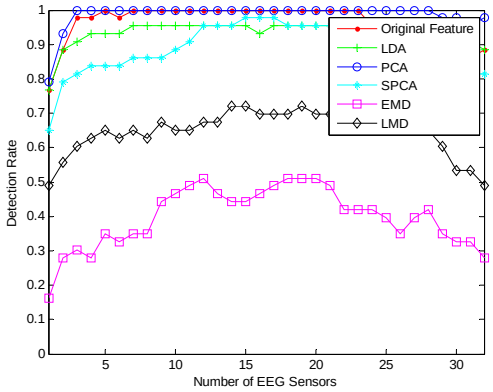
2) Chanel-wise LDA projection is the fastest method with acceptable accuracy. If Channel-wised PCA projection can not meet the real-time requirement, perhaps LDA is the only method we can use.



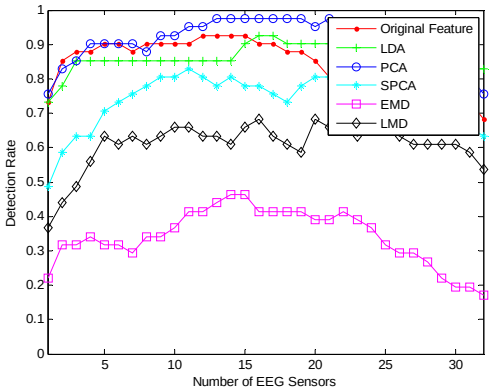

(a) Subject 1, Session 1.



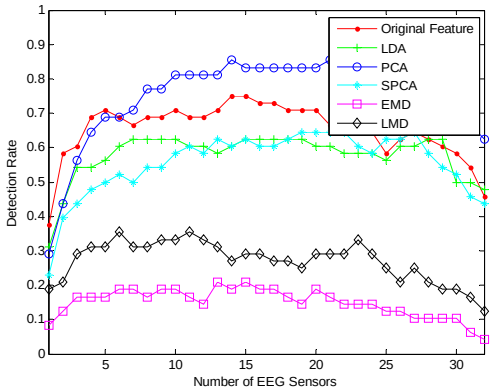

(b) Subject 1, Session 2



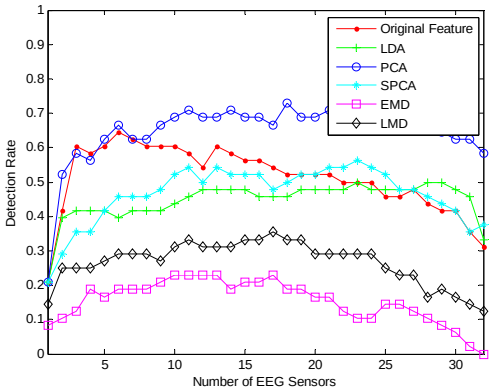

(c) Subject 2, Session 1.




(d) Subject 2, Session 2




(e) Subject 3, Session 1.




(f) Subject 3, Session 2

Figure 10-4 Feature ranking results for three subjects, in two sessions using different channel-wise dimensionality reduction and channel selection methods.

In theory, the properties of EMD and LMD suggest that they should be suitable for EEG data processing. However, neither method benefits our particular application. This illustrates that techniques used in the ERP detection is data dependent. In real application, cross validation process is preferred to select the optimal method.

From Chapter 7 to Chapter 10, we focused on another BCI application, single trial ERP detection. In Chapter 7, we introduced a baseline single trial ERP detection system. In Chapter 8, 9, and 10, we focused on different aspects in manipulating features. Experimental results showed that different methods are preferred for different applications and data. By carefully selecting dimensionality reduction methods, we can rank EEG features and do ERP detection in real-time system.

# Chapter 11: Conclusion

## 11.1 Summary

In this thesis, we focused on feature extraction, feature selection and dimensionality reduction techniques used in EEG based Brain Computer Interfaces (BCI).

In Chapter 1, we briefly reviewed the concept of BCI and its applications. EEG based BCIs have been widely used due to its non-invasive, low cost and high temporal resolution. However, motion artifacts, noise and the non-stationarity of brain EEG signals became major challenges to researchers. Furthermore, session-to-session transfer and subject-to-subject transfer are critical to the real-world BCI applications. Currently, a lot of research work has been done or are undertaking to solve these problems.

From the machine learning point of view, a BCI system can be treated as a classification system, which contains five parts: 1) signal pre-processing; 2) feature extraction; 3) feature selection and dimensionality reduction; 4) classification; and 5) post-processing. Theoretically, improving any part of BCI system can partly solve the challenges we mentioned above. However, getting the most compact and informative features, including feature extraction, feature selection and dimensionality reduction, not only makes the system robust, but also reduces the computational cost, which are the solutions to deal with the noise, realtime and nonstationarity problems. Therefore, we focused on feature manipulation. The thesis has two parts, corresponding to two different BCI applications: Augmented Cognition and single trial ERP detection.

### 11.1.1 Augmented Cognition

From Chapter 3 to Chapter 6, we focused on Augmented Cognition. In Chapter 3, we introduced the concept of Augmented Cognition. A baseline system was proposed to estimate cognitive mental states from human EEG signals when subjects were executing different mental tasks. Experimental results showed that the baseline system can classify two levels of workload even though the EEG signals are noisy and non-stationary. However, the classification accuracy is not satisfactory. From Chapter 4 to Chapter 6, we proposed different feature manipulation techniques to improve the performance of the system.

In Chapter 4, we described an ICA-MI framework to estimate mutual information between the features and the class labels. We used this framework to develop our feature selection and dimensionality reduction methods. We assumed that the linear assumption is satisfied and applied a linear ICA transformation on the EEG data. Experimental results showed that the system using our feature selection and dimensionality reduction approach performed better than our baseline system. Feature selection results also demonstrated our method's ability to partly solve the subject-to-subject and session-to-session transfer problems.

In some applications, the linear assumption may not hold, since we have no prior knowledge about the data. Therefore, in Chapter 5, we presented two non-linear feature selection methods: generalized ICA-MI methods and GMM-MI methods. The former one uses a piece-wise linear to approximate a nonlinear ICA transformation for mutual information estimation. The latter one uses Gaussian Mixture Model density estimation to estimate mutual information. Results from a synthetic dataset, UCI iris dataset, and from the AugCog dataset showed that the piece-wise linear ICA-MI feature selection method outperformed the linear ICA-MI method described in Chapter 4. Another advantage of the piece-wise linear ICA-MI method is that it can be treated as a generalized framework which contains linear ICA-MI framework as its special

case. However, this approximation requires that we have enough data samples within each partition; otherwise the projection error using the piece-wise linear ICA can be larger than that of the linear ICA.

The other non-linear feature selection method, GMM-MI, is more when there were not enough data samples for each partition. The GMM-MI method exploits the fact that GMM is a non-parametric method for density estimation and requires no prior knowledge about the data. The GMM-MI mutual information estimation framework was introduced, and a simplified GMM-MI feature selection algorithm was described in Section 5.2.2. Experimental results on the AugCog dataset, however, did not show that the GMM-MI performed better than the linear ICA-MI. Furthermore, the GMM-MI method is more computationally expensive than the ICA-MI method. However, when the linear assumption does not hold, and data size is not large enough to apply piece-wise linear ICA-MI method, GMM-MI maybe a viable alternative for feature selection.

Throughout Chapters 3 to 5, we used the Power Spectrum Density (PSD) features integrated over 5 well established frequency bands for each EEG channel, and did not use information about the correlation between different frequency bands. However, such correlation could offer additional information that can help extract intrinsic characteristics of brain dynamics in the frequency domain. In Chapter 6, we introduced an adaptive feature extraction method that was based on statistical similarity. Compared to the feature extraction method, which integrated frequency components using pre-defined frequency bands, the proposed method effectively discovered the correlation among different components. The integration process eliminated the redundant information and increased the Signal-to-Noise ratio. Experimental results showed that this method outperformed the feature extraction method used in the baseline system.

### 11.1.2 Single Trial ERP Detection

From Chapter 7 to Chapter 10, we focused on single trial ERP detection. In Chapter 7, we introduced the concept of single trial ERP detection and described an example of single trail ERP detection task under RSVP paradigm. A baseline system was proposed to solve the single trail ERP detection problem. We applied the baseline system on four day EEG data from four subjects. The experimental results were presented using ROC curves, the area under a ROC curve, and the minimum false alarm rate at zero missing (MFAR). Usually, in a classification system, area under a ROC curve corresponds to the classification accuracy: the larger the area, the more accurate the classifier is. However, in the single trial ERP detection task, since the data is very unbalanced, containing many more distracters than targets, using the area under ROC curve to evaluate the performance of ERP detector is inadequate. In some applications where the cost of missing a target is huge, the minimum false alarm rate at zero missing is also used as a criterion. In other applications, a high ERP detection rate with a reasonable false alarm rate is more desirable. The results in Section 7.4 illustrated that the overall area under ROC curves exceeded 90%. However, the overall MFAR was high as well. In the baseline system, we did not use much machine learning techniques to manipulate the features.

From Chapter 8 to Chapter 10, different feature extraction, feature selection and dimensionality methods were used to improve the performance. In Chapter 8, we described a temporal windowing scheme to divide the features in the time domain, classifying the features within each temporal window and fusing decision using Bayes method. We divided the [0-500ms] window into ten 50ms non-overlapping sub-windows. SVM classifiers were applied to the sub windows between 200ms and 450ms. Although

experiment results of the temporal windowing scheme did not exhibit better performance than the baseline system, it has several advantages. First, it utilizes a smaller dimensionality feature vector; thus is expected to be more robust over long-term BCI training. Using small dimensional features also contributes to the computational efficiency for realtime implementation. Second, by breaking the raw data into multiple discriminant temporal components, EEG channel selection becomes feasible for each sub window; this is important because at different phases of the response, different regions of the brain become active, therefore for each window, appropriate channels can be retained, contributing to further reduction of feature dimensionality, and hence classifier robustness. We may use the mutual information based EEG channel selection method described in Chapter 4 for each temporal sub window in order to achieve this goal. Third, the temporal window scheme could allow us to further improve the Bayesian fusion model utilizing more elaborate graphical models of dependencies between the temporal windows. Even with the naïve Bayesian fusion, this approach is competitive with our baseline approach in the experiments performed.

In Chapter 9, we explored the frequency and spatial domains of EEG signals, and compared different features extracted and selected from a range of frequency bands and EEG channel subsets. Experimental results indicated that the ERP energy is concentrated in the 0-1Hz and 10-20Hz bands. This suggested that not only P300, but also other ERPs, appear in our RSVP task. The results were consistent across subjects and sessions although we did not have enough subjects to employ statistical confidence tests. The performance of the error based feature ranking approach indicated that not all EEG channels were needed for ERP detection. Using a subset of EEG channels yielded better performance than using all channels. Furthermore, this subset was different across subjects. Channel ranking using the original features gave

accurate results, but it was computational intensive and not suitable for realtime system. One way to reduce the computational load was to apply the channel-wise dimensionality reduction before we ranked channels. LDA projection was used in this chapter. Although the projection sacrifices accuracy, the results were still better when channel selection was performed than using raw features without channel selection.

In Chapter 10, we compared different subspace projection methods for the channel-wise dimensionality reduction and feature selection. Experimental results showed that ERP detection using the original features and using PCA with the first 10 principal components of each channel performed the best, while the time-consuming SPCA, EMD, and LMD methods performed surprisingly worse, even for an offline system. The performance of LDA projection was acceptable and was the fastest method.

## 11.2 Discussion

In Chapter 1, we proposed that feature manipulation techniques can solve three problems we defined in Section 1.3: the robustness, the non-stationarity and realtime problems. We also proposed that BCI applications are data dependable and different feature manipulation techniques are desired for different mental tasks. The experimental results in this thesis validated our statements. A lot of future work can be done to further improve the performance of both systems; however, we must be aware of the current limitations. Knowing the current limitation of BCIs gives us a general concept of the application and performance of BCI systems.

First, BCI applications are based on our knowledge about the brain. Although neuroscientists and other brain related researchers have partly deciphered the relationship between neuron activities and human

cognitive states, full understanding of how the brain works is still not possible. This will greatly limit our applications of BCIs.

Second, although EEG based BCIs have a lot of advantages, as we described in Chapter 1, the signal quality is generally poor, which makes many real world applications infeasible. As the sensor techniques improve, we expect to acquire higher quality brain signals, and hence to broaden our BCI applications.

Third, we never knew the real brain activities of the subjects during mental tasks we described in Chapter 3 and Chapter 7. Irrelevant neural activities of the subjects would make it hard to find the true class labels. A classifier that is trained using incorrectly labeled data will adapt to a wrong decision. We could improve the experiment design and introduce other reference channels, such as eye trackers, or video cameras, to eliminate irrelevant data. However, introducing additional inputs will also increase the complexity of the system.

## 11.3 Future Work

Our future work will focus on the following aspects:

First, although we did comprehensive study on different feature manipulation techniques in BCI applications, a lot of work can still be done as the next step. For the ICA-MI framework, we used the generalized eigendecomposition to find the ICA solution and used the sample-spacings estimator to estimate single-dimensional entropy for their simplicity. Under the current framework, we can replace the ICA algorithm and the entropy estimator with any other solutions, so our study can focus on finding other methods to improve the performance of the ICA-MI feature selection method. In Chapter 6, we proposed a statistical similarity based feature extraction method, in which the frequency bands were partitioned using

the spectrum clustering algorithm and the segmented frequency components were integrated using a simple average. In the future, we will use the adaptive method in finding optimal weights to integrate frequency components within a partition. Similarly, we will study other feature manipulation techniques for single trial ERP detection.

Second, as shown in Figure 1-2, pre-processing is also a critical component in the classification system, especially in applications where subjects are in a mobile environment, where more motion artifacts are introduced, such as walking. We will focus our study on artifacts removal using different signal processing and machine learning techniques.

Third, in the discussion in Section 11.2, we mentioned that it is hard to know the real brain activities, and hence to get the true class labels. However, by carefully designing the experiments and introducing reference channels, such as eye trackers and video cameras, we can integrate additional information into the system, and combine them with EEG data to get the class labels that are as close to the truth as possible.

# References


Ai-ani, A., Deriche, M., “An Optimal Feature Selection Technique Using the Concept of Mutual Information,” Proceedings of ISSPA, pp. 477-480, 2001.

Anderer, P., *et al*., Artifact processing in computerized analysis of sleep EEG – a review. Neuropsychobiology 1999;40:150–157.

Anderson, C.W., Sijercic, Z., Classification of EEG signals from four subjects during five mental tasks. In *Solving Engineering Problems with Neural Networks: Proceedings of the International Conference on Engineering Applications of Neural Networks (EANN’96)*, 1996

Bach, F.R., Jordan, M.I., Learning spectral clustering. Neural Info. Processing Systems 16 (NIPS 2003), 2003.

Battiti, R., “Using Mutual Information for Selecting Features in Supervised Neural Networks learning,” IEEE Trans. Neural Networks, vol. 5, no. 4, pp. 537-550, 1994.

Beirlant, J., *et al*., Nonparametric entropy estimation, International J. Mathematical and Statistical Sciences, 6, pp. 17-39, 1997.

Belkin, M. Niyogi, P., Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, Advances in Neural Information Processing Systems 14 (NIPS 2001), pp: 585-591, MIT Press, Cambridge, 2002.

Berg, P., Scherg, M., A multiple source approach to the correction of eye artifacts. *Electroencephalography and Clinical Neurophysiology*, 90, 229-241, 1994.

Bishop, C.M., *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

Bostanov, V., 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 Transactions on Biomedical Engineering*, 51(6): 1057-1061, 2004.

Burges, C.J.C., "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery,* vol 2, pp.121-167, 1998.

Chapin, J.K., *et al*., Real-time control of a robot arm using simultaneously recorded neurons in the motor cortex. *Nat. Neurosci*. 2, 664-670, 1999.

Cincotti, F., *et al*., Comparison of different feature classifiers for brain computer interfaces. In *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, 2003.

Chow, T., Huang, D., Estimating optimal feature subsets using efficient estimation of high-dimensional mutual information. *IEEE Transaction Neural Network*, Vol. 16, No. 1, (January 2005) pp. (213-224), ISSN 1045-9227, 2005.

Cormen T.H., *et al*., "Introduction to Algorithms," MIT Press, pp. 441, 1990.

Cover, T.M., Thomas, J.A. *Information Theory*, Wiley, New York, 1991.

Croft, R.J., Barry, R.J., Removal of ocular artifact from the EEG: a review. Neurophysiol Clin 2000;30:5–19.

d'Aspremont, A., *et al*., A Direct Formulation for Sparse PCA Using Semidefinite Programming, SIAM Rev. Volume 49, Issue 3, pp. 434-448 (2007).

Dempster, A.P., *et al*., "Maximum Likelihood from Incomplete Data via the EM Algorithm," Journal of the Royal Statistical Society, vol. 39, pp. 1-38, 1977.

Devijver, P.A., Kittler, J., *Pattern Recognition: A Statistical Approach*, Prentice Hall, London, 1982.

Donchin E., Coles, M., “Is the P300 component a manifestation of context updating?” Behavioral Brain Science, 11, 357-374, 1988.

Donchin E, *et al*., The mental prosthesis: assessing the speed of a P300-based brain–computer interface. IEEE Trans Rehabil En g 2000;8:174–179.

Duda, R.O., *et al*., Pattern Classification, 2nd ed., Wiley, 2000.

Erdogmus, D, *et al.*, “Cognitive State Estimation Based on EEG for Augmented Cognition,” Proceedings of NER’05, pp. 566-569, Arlington, Virginia, Mar 2005.

Everson, R., Roberts, S., “Independent Component Analysis: A Flexible Nonlinearity and Decorrelating Manifold Approach,” Neural Computation, vol. 11, no. 8, pp. 1957-1983, 2003.

Fano, R. M., *Transmission of Information: A Statistical Theory of Communications*. Wiley, New York, 1961.

Farwell, L.A., Donchin, E, Talking off the top of your head: Toward a mental prosthesis utilizing event-related brain potentials. Electroencephalography and Clinical Neurophysiology, 70, page 512-523, 1988.

Fukunaga, K. *Introduction to Statistical Pattern Recognition*, 2nd ed., Academic Press, New York, 1990.

Gerson, A., *et al*., “Single-trial analysis of EEG for enabling cognitive user interfaces,” Ch. 40 in *Handbook of Neural Engineering*, M. Akay (Ed), Wiley/IEEE Press, 2006.

Gevins A., *et al*., “High Resolution EEG Mapping of Cortical Activation Related to Working Memory: Effects of Task Difficulty, Type of Processing, and Practice,” Cerebral Cortex, vol. 7, No. 4, pp. 374-385, 1997.

Gevins A., Smith M. E., Neurophysiological Measures of Working Memory and Individual Differences in Cognitive Ability and Cognitive Style. Cerebral Cortex, Vol. 10, No. 9, Sep. 2000.

Halgren, E., *et al*., Rapid Distributed Fronto-parieto-occipital Processing Stages During Working Memory in Humans. Cerebral Cortex, Vol. 12, No. 7, Jul. 2002.

Hagen, L., Kahng, A.B., New spectral methods for ratio cut partitioning and clustering. IEEE. Trans. on Computed Aided Desgin, 11:1074--1085, 1992.

Haykin, S., Neural Networks – A comprehensive Foundation, 2nd ed. Englewood Cliffs, NJ, Prentice-Hall, 1998.

Hellman, M.E., Raviv, J., "Probability of Error, Equivocation and the Chernoff Bound," IEEE Transactions on Information Theory, vol. 16, pp. 368-372, 1970.

Hild, K.E. *et al.*, "Blind Source Separation Using Renyi's Mutual Information," IEEE Signal Processing Letters, vol. 8, no. 6, pp. 174-176, 2001.

Hild, K.E., *et al.*, Feature extraction using information-theoretic learning. *IEEE Transaction Pattern Analysis and MachineIntelligence*, Vol. 28, No. 9, (September 2006) pp. (1385-1393), ISSN 0162-8828, 2006.

Hillyard, S.A., Galambos, R., Eye-movement artifact in the CNV. *Electroencephalographic Clinical Neurophysiology* 28(2):173-82, 1970.

Hoffer, J. A., *et al*., Neural signals for command control and feedback in functional neuromuscular stimulation: a review. *J Rehabil Res Dev, 33,* 145-157, 1996.

Huang, N., et al., "The empirical mode decomposition, and Hilbert spectrum for nonlinear and non-stationary time series analysis," *Proc.R. Soc. London* **454**, 903–995, 1998.

Huang, Y., *et al*., "Comparison of linear and nonlinear approaches in single trial ERP detection in rapid serial visual presentation tasks," in *Proceedings of IEEE International Joint Conference on Neural Networks***,** Vancouver, Canada, 2006a.

Huang, Y., *et al*., "Boosting linear logistic regression for single trial ERP detection in rapid serial visual presentation tasks," in *Proceedings of the 28th International Conference of IEEE EMBS*, New York, 2006b.

Huang, Y., *et al.*, Large-scale image database triage via EEG evoked responses, in *Proceedings of the 2008 IEEE International Conference of Acoustics, Speech, and Signal Processing***,** Las Vegas, 2008.

Hyvärinen, A., *et al.*, "Image Feature Extraction by Sparse Coding and Icomponent Analysis," Proceedings of ICPR'98, pp. 1268-1273, 1998.

Hyvarinen A, Pajunen P. Nonlinear independent component analysis: Existence and uniqueness results, Neural Netw. 1999 Apr;12(3):429-439.

Hyvärinen, A., Oja, E., "A Fast Fixed Point Algorithm for Independent Component Analysis", Neural Computation, vol. 9, no. 7, pp. 1483-1492, 1997.

Jervis, B.W., *et al*., The removal of ocular artifacts from the electroencephalogram: a review. *Medical and Biological Engineering and Computing*, 26, 2-12, 1988.

Johnson, R., The amplitude of the P300 component of the event-related potential, Advances in Psychophysiology, Vol2, pp69-138, 1988.

Jung, T.P., *et al*., Removing electroencephalographic artifacts by blind source separation. Psychophysiology 2000;37:163–178.

Karhunen, J., *et al.*, “Local linear independent component analysis based on clustering”. International Journal of Neural Systems, vol.10, no. 6, 2000, pp. 439-451.

Kilgore, K. L., *et al.*, An implantable upper-extremity neuroprothesis. *J Bone Joint Surg, 79A*, 533-541, 1997.

Kruse A., Schmorrow, D., Foundations of Augmented Cognition. 11th International Conference on Human-Computer Interaction, vol 11, 2005.

Krusienski,D.J., *et al.*, "A comparison of classification techniques for the P300 Speller" Journal of Neural Engineering vol.3, pp.299-305 (2006).

Krystal A., *et al.*, “New methods of time series analysis for nonstationary EEG data: eigenstructure decompositions of time varying autoregressions”. Clin. Neurophysiol. v110. 1-10, 1999.

Kwak, N., Choi, C-H., “Input Feature Selection for Classification Problems,” IEEE Transactions on Neural Networks, vol. 13, no. 1, pp. 143-159, 2002.

Lan, T., *et al.*, "Estimating Cognitive State Using EEG Signals," Proceedings of EUSIPCO’05, Antalya, Turkey, Sep 2005a.

Lan, T., *et al.*, "Salient EEG Channel Selection in Brain Computer Interfaces by Mutual Information Maximization," Proceedings of EMBC’05, pp. 7064 – 7067, Shanghai, China, Sep 2005b.

Lan, T., *et al.*, "Feature Selection by Independent Component Analysis and Mutual Information Maximization in EEG Signal Classification," Proceedings of IJCNN’05, vol. 5, pp. 3011-3016, Montreal, Quebec, Aug 2005c.

Lan, T., Erdogmus, D., "Local Linear ICA for Mutual Information Estimation in Feature Selection," Proceedings of MLSP’05, pp. 3-8, Mystic, Connecticut, Sep 2005.

Lan, T., *et al*., "A Comparison of Linear ICA and Local Linear ICA for Mutual Information Based Featue Ranking," Proceedings of ICA 2006, pp. 823-830, Charleston, South Carolina, USA, Mar 2006a.

Lan, T., et al., "Estimating Mutual Information Using Gaussian Mixture Models for Ranking and Selection," Proceedings of IJCNN 2006, pp. 5034 – 5039, Vancouver, July, 2006b.

Lan, T., *et al*., "Automatic Frequency Bands Segmentation using Statistical Similarity for Power Spectrum Density based Brain Computer Interfaces," Proceedings of IJCNN 2006, pp. 4650 – 4655, Vancouver, July, 2006c.

Lan, T., *et al.*, "Feature and Channel Selection for Cognitive State Estimation using Ambulatory EEG," Journal of Computational Intelligence and Neuroscience, vol. 2007, Article ID 74895, 12 pages, Sep 2007. (Special Issue on Brain-Computer Interfaces).

Lan, T, Erdogmus, D., "Maximally Informative Feature and Sensor Selection in Pattern Recognition using Local and Global Independent Component Analysis," Journal of VLSI Signal Processing Systems, vol.48, no. 1-2, pp. 39-52, Aug.2007 (invited paper for the Special Issue of Selected MLSP 2005 Papers).

Lan, T., *et al*., "A Comparison of Temporal Windowing Schemes for Single-trial ERP Detection," Proceedings of NER 2009, Antalya, Turkey, Apr 2009.

Lan, T., *et al*., "Identifying informative features for ERP speller systems based on RSVP paradigm," ESANN 2010 proceedings, pp.351-356, Belgium, Apr, 2010a.

Lan, T., *et al*., "A comparison of different dimensionality reduction and feature selection methods for single trial ERP detection," Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society (2010), pp. 6329-6332, August, 2010b.

Learned-Miller, E.G., Fisher III, J.W., “ICA Using Spacings Estimates of Entropy,”.Journal of Machine Learning Research, vol. 4, pp. 1271-1295, 2003.

Lemm, S., *et al*., “Spatio-spectral filters for improving the classification of single trial EEG,” *IEEE Transactions on Biomedical Engineering*, 52(9):1541–1548, 2005.

Li, R., Principe J.C., “Blinking Artificat Removal in Cognitive EEG Data Using ICA”, Proceedings of the 28th IEEE EMBS Annual International Conference, Aug. 30, 2006, pp. 5273-5276.

McFarland, D.J., *et al*., Spatial filter selection for EEG-based communication. Electroenceph clin Neurophysiol 1997;103:386–394.

Middendorf, M., *et al*., “Brain–computer interfaces based on steady-state visual evoked response”, IEEE Trans Rehabil Eng , vol.. 2, pp. 211–213, 2008.

Muller-Gerking J, *et al*., Designing optimal spatial filters for single-trial EEG classification in a movement task. Clin Neurophysiol 1999;110:787–798.

Murillo, J., Rodriguez, A., Maximization of mutual information for supervised linear feature extraction. *IEEE Transaction Neural Network*, Vol. 18, No. 5, (September 2007) pp. (1433-1441), ISSN 1045-9227, 2007.

Nunez, P.L., *et al*., A theoretical and experimental study of high resolution EEG based on surface Laplacians and cortical imaging. Electroenceph clin Neurophysiol 1994;90:40–57.

Nunez, P.L., *et al*., EEG coherency. I: Statistics, reference electrode, volume conduction, Laplacians, cortical imaging, and interpretation at miltiple scales. Electroenceph clin Neurophysiol 1997;103:499–515.

Obermaier, B., *et al*., "Hidden markov models for online classification of single trial EEG data," *Pattern Recognit. Lett.*, vol. 22, pp. 1299–1309, 2001.

Oja, E., *Subspace Methods of Pattern Recognition*, Wiley, New York, 1983.

Palaniappan, R., Brain computer interface design using band powers extracted during mental tasks. In *Proceedings of the 2nd International IEEE EMBS Conference on Neural Engineering*, 2005.

Parra, L., Sajda, P., "Blind Source Separation via Generalized Eigenvalue Decomposition", Journal of Machine Learning Research, vol. 4, pp. 1261-1269, 2003.

Parra, L.C., *et al*., "Recipes for the linear analysis of EEG," *Neuroimage*, 28(2), 326-341, 2005.

Parra, L.C., *et al*., "Spatiotemporal Linear Decoding of Brain State", in *IEEE Signal Processing Magazine*, Jan. 2008, pp.107-115.

Parzen, E., "On Estimation of a Probability Density Function and Mode", in Time Series Analysis Papers, Holden-Day, Inc., San Diego, California, 1967.

Pavel, M, *et al*., Augmented cognition: Allocation of attention, Proceedings of 36th Hawaii International Conference on System Sciences January 6-9, 2003, Big Island, HI, USA. IEEE Computer Society, 2003.

Pfurtscheller, G., EEG event-related desynchronization (erd) and event-related synchronization (ers), 1999.

Pregenzer, M., Pfurtscheller, G., "Frequency Component Selection for an EEG-Based Brain to Computer Interface," IEEE Transactions on Rehabilitation Engineering, vol. 7, no. 4, 1999.

Rabiner, L.R., A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, page 257-286, 1989.

Russell C.A., Gustafson S.G., “Selecting Salient Features of Psychophysiological Measures,” Air Force Research Laboratory Technical Report (AFRL-HE-WP-TR-2001-0136), 2001.

Sajda, P., *et al*., “Single-trial analysis of EEG during rapid visual discrimination: Enabling cortically-coupled computer vision,” in *Towards Brain-Computer Interfacing*, Eds. G. Dornhege, J. R. Millan, T. Hinterberger, D.J. McFarland and K.R. Muller. MIT Press, invited, in press, 2007.

Schalk, G, *et al*., EEG-based communication and control: presence of error potentials. Clin Neurophysiol 2000;111:2138–2144.

Schiff S.J., *et al*., “Fast wavelet transform of EEG. Electroenceph”. Clin. Neurophysiol., 1994, 91, 442-455.

Schmorrow, D. D., Kruse, A. A. Augmented Cognition. In W.S. Bainbridge (Ed.), Berkshire Encyclopedia of Human-Computer Interaction (pp. 54-59). Great Barrington, MA: Berkshire Publishing Group, 2004.

Smith, J. S., The local mean decomposition and its application to eeg perception data, Journal of The Royal Society Interface . (2005) 443–454.

Sperlich F-J, Hillyard SA, “Intra-modal and cross-modal spatial attention to auditory and visual stimuli. An event related brain potential study”. Cogn Brain Res, vol. 2, pp. 327–343, 2004.

Szummer, M., Jaakkola, T., “Information Regularization with Partially Labeled Data”, Advances in NIPS 15, 2002.

Thakor N. V., *et al*., “Multiresolution wavelet analysis of evoked potentials”. IEEE Trans. on Biomed. Engin., 1993, 11, 1085-1093.

Thorpe, S., *et al*., Speed of processing in the human visual system, Nature, 381, 520-522, 1996.

Torkkola, K., “Feature Extraction by Non-Parametric Mutual Information Maximization,” Journal of Machine Learning Research, vol. 3, pp. 1415-1438, 2003.

Vidal, J., "Toward Direct Brain–Computer Communication", in Annual Review of Biophysics and Bioengineering, L.J. Mullins, Ed., Annual Reviews, Inc., Palo Alto, Vol. 2, 1973, pp. 157-180.

Vidal, J., "Real-Time Detection of Brain Events in EEG", in IEEE Proceedings, May 1977, 65-5:633-641.

Welch, P., “The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short Modified Periodograms”, IEEE Transactions on Audio and Electroacoustics, vol. 15, no. 2, pp. 70-73, 1967.

Whitton J.L., *et al*., A spectral method for removing eye-movement artifacts from the EEG. *Electroencephalographic Clinical Neurophysiology*;44:735-41, 1978.

Widrow, B., Hoff, M. E., "Adaptive switching circuits," in IRE WESCON Convention Record, 1960, pp. 96-104.

Wolpaw, J.R., McFarland, D.J., “Multichannel EEG-based brain-computer communication”, *Electroencephalogr. Clin. Neurophysiol.* 1994, 90 444–9.

Wolpaw, J.R., McFarland, D.J., “Control of a two-dimensional movement signal by a noninvasive brain-computer interface in humans” *Proc. Natl Acad. Sci.* 2004, 101 17849–54.

Wolpaw, J.R., *et al*., An EEG-based brain computer interface for cursor control. *Electroencephalogr Clin Neurophy*, 78, 252–259, 1991.

Yang, H.H., Moody, J., “Feature Selection Based on Joint Mutual Information,” in Advances in Intelligent Data Analysis and Computational Intelligent Methods and Application, 1999.

Yang, H.H., Moody, J., “Data Visualization and Feature Selection: New Algorithms for Nongaussian Data,” Advances in NIPS, pp. 687-693, 2000.

Zha, H., *et al*., Spectral relaxation for K-means clustering. Advances in Neural Information Processing Systems 14 (NIPS 2001). pp. 1057-1064, Vancouver, Canada. Dec. 2001.