

Washington University in St. Louis
Washington University Open Scholarship

All Theses and Dissertations (ETDs)

January 2011

Decoding Brain Activation from Ipsilateral Cortex using ECoG Signals in Humans

Yuzong Liu

Washington University in St. Louis

Follow this and additional works at: <https://openscholarship.wustl.edu/etd>

Recommended Citation

Liu, Yuzong, "Decoding Brain Activation from Ipsilateral Cortex using ECoG Signals in Humans" (2011). *All Theses and Dissertations (ETDs)*. 513.

<https://openscholarship.wustl.edu/etd/513>

This Thesis is brought to you for free and open access by Washington University Open Scholarship. It has been accepted for inclusion in All Theses and Dissertations (ETDs) by an authorized administrator of Washington University Open Scholarship. For more information, please contact digital@wumail.wustl.edu.

WASHINGTON UNIVERSITY IN ST. LOUIS
School of Engineering and Applied Science
Department of Computer Science and Engineering

Thesis Examination Committee:
Professor Kilian Weinberger, Chair
Professor Robert Pless
Professor Viktor Gruev

DECODING BRAIN ACTIVATION FROM IPSILATERAL CORTEX USING
ECOG SIGNALS IN HUMANS

by

Yuzong Liu

A thesis presented to the School of Engineering
of Washington University in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

May 2011
Saint Louis, Missouri

ABSTRACT OF THE THESIS

Decoding Brain Activation from Ipsilateral Cortex using ECoG Signals in Humans

by

Yuzong Liu

Master of Science in Computer Science

Washington University in St. Louis, 2011

Research Advisor: Professor Kilian Weinberger

Today, learning from the brain is the most challenging issue in many areas. Neural scientists, computer scientists, and engineers are collaborating in this broad research area. With better techniques, we can extract the brain signals by either non-invasive approach such as EEG (electroencephalography), fMRI, or invasive method such as ECoG (electrocorticography), FP (field potential) and signals from single unit. The challenge is, given the brain signals, how can we possibly decipher them?

Brain Computer Interfaces, or BCIs, aim at utilizing the brain signals to control prosthetic arms or operate devices. Previously almost all the research on BCIs focuses on decoding signals from the contralateral hemisphere to implement BCI systems. However, the loss of functionality in the contralateral cortex often occurs due to strokes, resulting in total failure to motor function of fingers, hands, and limbs contralateral to the damaged hemisphere. Recent studies indicate that the signals from ipsilateral cortex is relevant to the planning phase of motor movements. Therefore, it is critical to find out if human motor movements can be decoded using signals from

the ipsilateral cortex. In the thesis, we propose using ECoG signals from the ipsilateral cortex to decode finger movements. To our knowledge, this is the first work that successfully detects finger movements using signals from the ipsilateral cortex. We also investigate the experiment design and decoding directional movements. Our results show high decoding performance. We also show the anatomical feature analysis for ipsilateral cortex in performing motor-associated tasks, and the features are consistent with previous findings. The result reveals promising implications for a stroke relevant BCI.

Acknowledgments

It is a memorable two-year time living in St. Louis, and I will always cherish the time here in Washington University: I would like to thank my advisor, Professor Kilian Weinberger, for having me study in the machine learning group. He truly shed light on me in machine learning area and I learned a lot from him in research, in lectures, and in seminars. Also, I would like to thank Professor Eric Leuthardt for having me involved in the emerging new area of brain machine interfaces. I would like to thank Professor Robert Pless and Professor Viktor Gruev for serving as my thesis committee members. I would like to thank all the PhD students at the Machine Learning Group: Eddie Xu, Minmin Chen, Stephen Tyree and Dor Kedem. Many thanks to my friends here in St. Louis, and special thanks to my office mate Rahav Dor. Finally, I would love to thank my family for their unconditional love and support.

Yuzong Liu

Washington University in Saint Louis
May 2011

Dedicated to my family, for their unconditional support and love.

Contents

Abstract	ii
Acknowledgments	iv
List of Tables	viii
List of Figures	ix
1 Background of Brain Computer Interfaces and ECoG Signals . . .	1
1.1 Background of BCI	1
1.2 Survey of Current BCI Signal Modalities and Electrocorticography(ECoG) Signals	2
1.3 Why Decode from Ipsilateral Cortex?	3
1.4 Thesis Outline	5
2 Experiment Setup	7
2.1 Subjects	7
2.2 Experiment Paradigm and Data Acquisition	8
3 ECoG Signal Processing	10
3.1 Gabor Filter Analysis	10
3.2 Dimensionality Reduction	12
3.3 Electrode Co-Registration	14
4 Machine Learning Algorithms	15
4.1 Notations	15
4.2 ℓ_1 -Regularized Logistic Regression	16
4.2.1 Logistic Regression for Binary Classification	16
4.2.2 Overfitting and Regularization	17
4.2.3 ℓ_1 norm vs ℓ_2 norm	18
4.3 Support Vector Machines	18
4.4 Kernel SVM	22
4.5 Multiclass Classification	24
4.6 LSR with Min-Max Constraint	24
4.7 Multitask Learning	25
4.7.1 Introduction to Multitask Learning	25
4.7.2 SVM-MTL and LR-MTL	28

5	Results	29
5.1	Evaluation Metric	29
5.2	Time Lag	29
5.3	Detecting Finger Movement	32
5.4	Learning Commonality of the Brain Activity	32
5.5	Feature Analysis	33
5.6	Kinematic Decoding	34
6	Additional Work	39
6.1	Experiment Setup for Directional Movement Classification	39
6.2	Classification between Resting and Movement State	40
6.3	Classification between Directions	41
7	Discussion	44
8	Conclusion	46
	References	47
	Vita	50

List of Tables

1.1	Comparison of Different signals for Brain Computer Interfaces	2
2.1	The information of patients and the information of implanted electrode grids.	7
5.1	Multitask learning vs single-task learning for Subject 1.	33
5.2	Multitask learning vs single-task learning for Subject 2.	33
5.3	Multitask learning vs single-task learning for Subject 3.	34
5.4	Prediction the flexion of finger movements.	35

List of Figures

1.1	Contralateral and Ipsilateral Cortex. For contralateral movement, the left hemisphere controls the right hand, and vice versa. For ipsilateral movement, the left hemisphere controls the left hand, and vice versa.	3
1.2	Damage in One Hemisphere. There is a hemispherical damage to the left cortex, resulting in inability to control the right limb using signals from contralateral cortex.	4
2.1	Schematic diagram of the experiment setup.	9
3.1	Example of PCA for 2-D data.	13
4.1	Logistic function. With input value z on the x-axis, and output value $f(z)$ on the y-axis.	16
4.2	Different norms of regularization term. ℓ_1 norm treats each feature equally the same; while ℓ_2 norm gives more penalty on the features with larger weights.	19
4.3	Example of SVM in 2D space. The circles are the positive examples, and squares are the negative examples. The examples with double stroke lines around are the ones that violate the strict constraint of linear SVM.	20
4.4	Single-task learning for detecting finger movements.	27
4.5	Multi-task learning for detecting finger movements. The green circle is the shared representation \mathbf{w}_0 of all fingers' movements, and the red circles $\mathbf{w}_i, i = 1, 2, \dots, k$ are the specific representation of finger i . The prediction at time t for finger k is defined as $y_t = (\mathbf{w}_0 + \mathbf{w}_k)^\top \vec{x}_t$	27
5.1	Receiver Operating Characteristics. X-axis is the false positive rate, and y-axis is the true positive rate.	30
5.2	Different time-lag and its corresponding decoding performance. X-axis is the value of time-lag (seconds), and y-axis is the AUC.	31
5.3	ROC curve for the ipsilateral finger movement decoder. Horizontal axis shows the false positive rate, and the vertical axis shows the true positive rate. The dotted line is the accuracy of a random classifier. Classifiers that have higher <i>area under the ROC curve</i> , or AUC, indicate better classification performance.	36
5.4	The area under the curve (AUC) as a function of the number of features used for classification. Features were selected in decreasing order of their respective absolute weights from logistic regression with ℓ_1 -regularization.	37
5.5	μ -ECoG technology: implantation of micro-electrode on human brains.	37
5.6	Brain map representing the weights of the top 30 features of the three patients.	38

5.7	Selected prediction of the flexion of finger movements.	38
6.1	The experiment paradigm for directional movement classification.	40
6.2	The experiment test state.	41
6.3	The ROC curve of the classification.	42
6.4	Predicted movement versus the actual movement.	42
6.5	Classification between directions.	43

Chapter 1

Background of Brain Computer Interfaces and ECoG Signals

1.1 Background of BCI

The new century has witnessed a remarkable stride in advancing neuro-engineering technology. Brain computer interfaces (BCI), or brain machine interface (BMI), is a direct method to communicate between the brain and external media such as robotic limbs, and computer programs. The emerging new area of BCI allows patients who suffer from epilepsy, stroke, or loss of limbs due to accidents, warfare a new approach to regain their motor-associated function, by controlling their limbs (or artificial limbs) via brain computer interfaces. A robust BCI system should be able to directly interact with the brains by implanting electrodes on neural tissues, sensing the brain activity, translating the activity into commands for prosthetic arms and computers. Machine learning approach in BCI research has emerged in the past decades. Inherently, machine learning is ideal approach in the sense that we want to model how human brains are learning.

Signal	Regional Domain	Neuron Population	Invasiveness
EEG	3-5 cm	N/A	Non-invasive
ECoG	0.5 - 1 cm	large	Invasive
Field Potential	1 mm	large	Invasive
Single Unit	200 μ m	single neuron	Invasive

Table 1.1: Comparison of Different signals for Brain Computer Interfaces

1.2 Survey of Current BCI Signal Modalities and Electrocorticography(ECoG) Signals

There exists several signal modalities for the Brain Computer Interfaces research work. Electroencephalography signal, or EEG signal in short, records the electric activity from the scalp produced by the firing of neurons within the brain. Field potential, or FP, records the firing information of population neurons. Signals from single unit records the information from single neuron. Electrocorticography (ECoG), or signal recorded from the surface of the brain, offers an excellent opportunity to further define what level of motor information can be deciphered from human ipsilateral cortex related to movements (e.g. gross motor movements versus fine motor kinematics of individual finger movements). Here, we list four most widely used signal modalities for BCI in Table 1.1. In particular, we listed the regional domain, neuron population, and their corresponding surgical invasiveness.

In practice, most BCI system use EEG signals and ECoG signals, due to the non-invasiveness of EEG signal, and surgical feasibility of ECoG signal. The ECoG signal is more robust compared to the EEG signal: its magnitude is typically five times larger, its spatial resolution as it relates to independent signals is much greater (0.125 versus 3.0 cm for EEG), and its frequency bandwidth is significantly higher (0-550 Hz versus 0- 40 Hz for EEG) [12, 27]. When analyzed on a functional level, many studies have revealed that different frequency bandwidths carry highly specific and anatomically focal information about cortical processings.

The lower-frequency bands known as mu (8-12 Hz) and beta (18-26 Hz) are thought to be produced by thalamocortical circuits and often decrease in amplitude in association

with actual or imagined movements [13, 23]. Higher frequencies (> 30 Hz), or gamma rhythms, are thought to be produced by smaller cortical assemblies and have been associated with numerous aspects of speech and motor function [8, 17]. These higher frequency rhythms are only accessible through ECoG. Thus far, however, no studies have utilized these ECoG spectral features to definitively analyze and decode cortical processing of the specific kinematics of ipsilateral finger movements.

1.3 Why Decode from Ipsilateral Cortex?

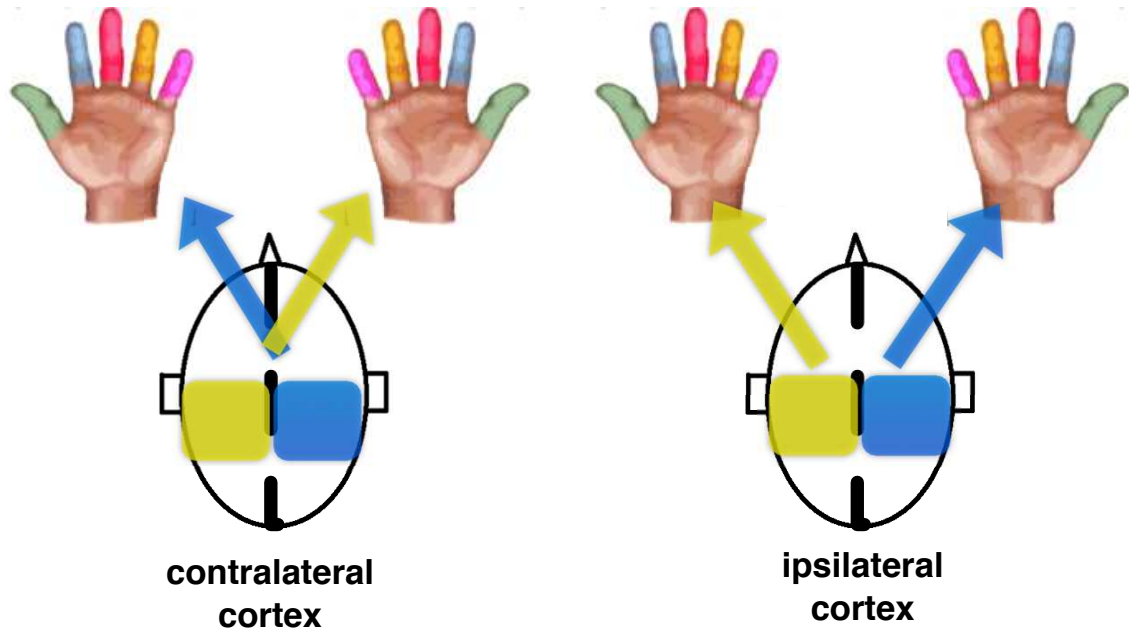


Figure 1.1: Contralateral and Ipsilateral Cortex. For contralateral movement, the left hemisphere controls the right hand, and vice versa. For ipsilateral movement, the left hemisphere controls the left hand, and vice versa.

Up to now, contralateral cortex is the key to human motor function, whereas ipsilateral cortex functions as the planning part of the motor control. For example, the left hand movement is controlled by the right hemisphere of the brain, and vice versa (Figure 1.1). So far all the research of motor control is based on using signals from the contralateral cortex of the brain.

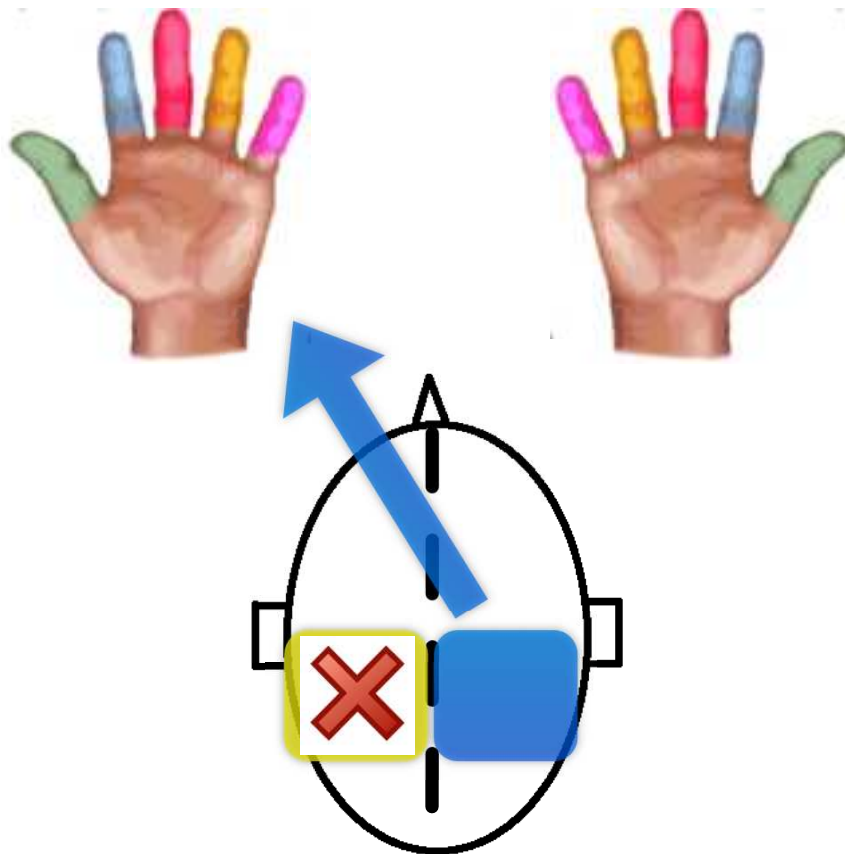


Figure 1.2: Damage in One Hemisphere. There is a hemispherical damage to the left cortex, resulting in inability to control the right limb using signals from contralateral cortex.

In this work, we will try to build a BCI system that detects finger movements using signals from ipsilateral cortex of the brain. The significance of this experimental paradigm lies in three aspects. In the first place, this is extremely important when the patients suffer from hemispheric dysfunction due to stroke or trauma. As indicated in Figure 1.2, if one hemisphere is damaged by epilepsy or stroke, a traditional BCI system will not be able to control the limb contralateral to the damaged hemisphere. Second, if a BCI system can accurately decode the brain activation, the potential surgical hazard will be alleviated. Only one hemisphere will be covered by the electrode grid, rather than both hemispheres. Thus, the patients will have less health hazard. Third, the ipsilateral cortex shows increased activities in the premotor regions rather

than the motor regions, and is believed to play the planning role in human movements. However, the exact role of ipsilateral cortex is still uncertain when it comes to human motor-related tasks. If a BCI system can use ipsilateral brain signal to perform the decoding, we can find a bypass to control the limbs without using the contralateral motor cortex.

This results in the growing need of decoding brain activation from the ipsilateral cortex. Thus far, however, no studies have utilized these ECoG spectral features to definitively analyze and decode cortical processing of the specific kinematics of ipsilateral finger movement. Until more recently, the first demonstration of this concept of utilizing ipsilateral motor signals for simple device control have been published both with ECoG (in healthy subjects) and MEG (in stroke patients) [2, 30]. In this study we set out to further explore the decoding of individual finger movements of the ipsilateral hand that could potentially be utilized for more sophisticated BCIs in the future.

1.4 Thesis Outline

In Chapter 1, we give a brief background of the emerging new area on brain computer interfaces. Different signal modality in BCI research is compared, and we stress the important of decoding brain activation from the ipsilateral cortex in humans.

In Chapter 2, we show the research subject information and the experiment setup.

In Chapter 3, we show the procedures and methods to process the ECoG signals. They include the idea of using Gabor Filter as bandpass filter, principle component analysis as a method to reduce the dimensionality and electrode co-registration.

In Chapter 4, we go over the machine learning algorithms in our work. For detecting finger movements, we show two classic methods: logistic regression, and support vector machines. We also discuss how to deal with overfitting using regularization, and discuss kernel methods in SVMs. For feature selection, we show the ℓ_1 norm gives the sparse representation of features. For kinematic decoding, we show a modified

LSR model. And we also bring the idea of multitask learning into the finger movement classification tasks.

In Chapter 5, we show the result of the finger movement tasks and they include: the effect of time-lag, detection of finger movement, multitask learning, prediction of finger positions, and feature analysis.

In Chapter 6, we show another scenario in BCI research: directional movement classification. We show the result of discriminating between resting and movement states, and discrimination between different direction targets.

In Chapter 7 and Chapter 8, we discuss our experimental results and conclusion of the work.

Chapter 2

Experiment Setup

2.1 Subjects

In our work, we have three patients for the experiment, all of whom gave informed consent. The subjects in this study were three patients (females; 8, 36, 48 years of age) with intractable epilepsy who underwent temporary placement of intracranial electrode arrays to localize seizure foci prior to surgical resection. All had normal levels of cognitive function and all were right-handed. The information of the three patients is listed in Table 2.1. The study was approved by the Washington University Human Research Protection Office.

Subject ID	Age	Gender	Grid Size	Grid Location
1	8	F	8×8	Right Hemisphere
2	36	F	8×10	Left Hemisphere
3	48	F	8×10	Left Hemisphere

Table 2.1: The information of patients and the information of implanted electrode grids.

2.2 Experiment Paradigm and Data Acquisition

The experiment paradigm is shown in Figure 2.1. The information of the finger movements was collected by a data glove (produced by Fifth Dimension, Irvine, CA), which gave the precise information on position, velocity, and acceleration. Each patient was seated in a hospital bed in front of a 17-inch LCD video monitor. The monitor gave the patient cues about which finger to move randomly, and the patients would follow the cue to move the corresponding finger at a self-paced speed until the cue disappeared from the monitor. Each cue lasts 1.5 seconds - 2.5 seconds, and on average each patient was given 30 random cues. The ECoG signals were recorded by the implanted electrode grids (8×8 , and 8×10). The ECoG signal were initially processed by BCI 2000 ([26]). Here, the general task in our work is to find a mapping from the brain activation, in this scenario the ECoG signals from the implanted electrode grids, to the finger movement.

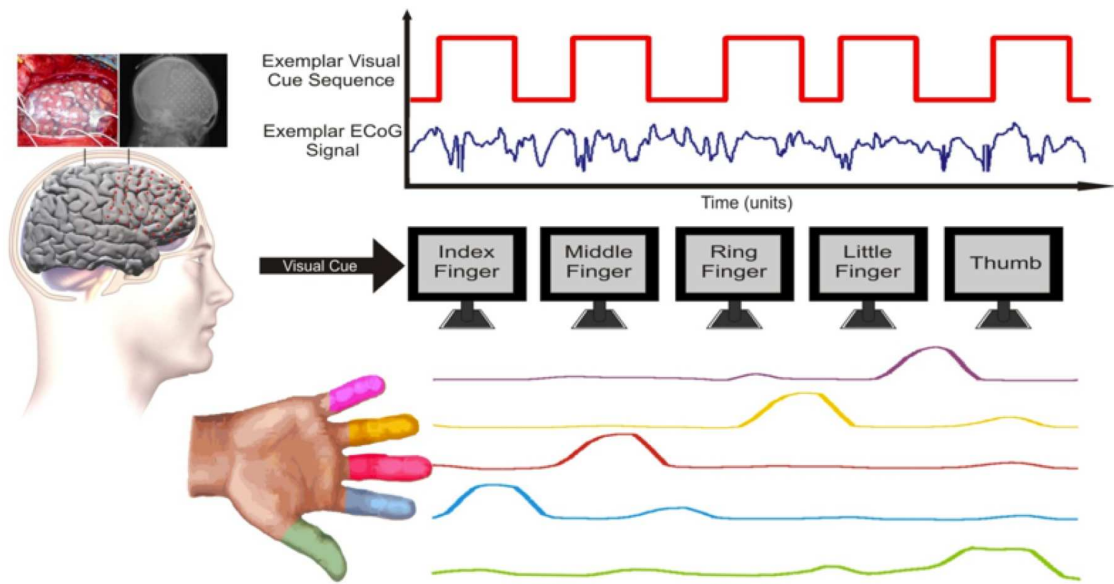


Figure 2.1: Schematic diagram of the experiment setup.

Chapter 3

ECoG Signal Processing

3.1 Gabor Filter Analysis

In signal processing, time-frequency analysis is important and we often convert the time-domain signal $g(t)$ to a proper representation domain $\hat{g}(f, t)$, where $\hat{g}(f, t)$ is the amplitude of frequency f at time t . Existing approaches such as Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), and more recently, wavelet transform are used to do the time-frequency analysis. Gabor filter is a popular approach to design band-pass filter. In 1-D space, such as speech signals, brain signals, it is referred as temporal Gabor filter; in 2-D space, such as images, it is referred as spacial Gabor filter. Here we follow the definition from Movellan [19], and write the Gabor filter as the product of a Gaussian kernel multiplied by a complex sinusoid $s(t)$:

$$g(t) = ke^{j\theta}w(at)s(t) \quad (3.1)$$

where

$$w(t) = e^{-\pi t^2} \quad (3.2)$$

$$s(t) = e^{j(2\pi f_0 t)} \quad (3.3)$$

$$e^{j\theta}s(t) = e^{j(2\pi f_0 t + \theta)} = (\sin(2\pi f_0 t + \theta), j \cos(2\pi f_0 t + \theta)) \quad (3.4)$$

Then we take the Fourier transform of Equation 3.1, i.e.

$$\hat{g}(f) = ke^{j\theta} \int_{-\infty}^{\infty} e^{-j2\pi ft} w(at) s(t) dt = ke^{j\theta} \int_{-\infty}^{\infty} e^{-j2\pi(f-f_0)t} w(at) dt = \frac{k}{a} e^{j\theta} \hat{w}\left(\frac{f-f_0}{a}\right) \quad (3.5)$$

where

$$\hat{w}(f) = w(f) = e^{-\pi f^2} \quad (3.6)$$

In this work, power spectrum of the Gabor filter output is used. According to Equation 3.5, the real part and imaginary part of the Gabor filter is phase-sensitive. To make it insensitive to phase θ , we use the power spectrum of the Gabor filter, given as:

$$\|g(f)\| = \frac{k}{a} \hat{w}\left(\frac{f-f_0}{a}\right) \quad (3.7)$$

Here f_0, a are the parameters of the Gabor filter, which define the centered frequency of Gaussian function, and the width of the Gabor filter at frequency f .

Parameters of the Gabor Filter: f_0 and a

As shown in Equation 3.7, f_0 is the peak response of the band-pass filter. a is the window size of the Gabor filter. To have better mathematical intuition, let us define the half-magnitude bandwidth as δ_f . We have the following equation

$$\hat{w}\left(\frac{f-f_0}{a}\right) = e^{-\pi\left(\frac{f-f_0}{a}\right)^2} = 0.5 \quad (3.8)$$

The half-magnitude bandwidth is calculated as

$$\delta_f = 2(f-f_0) = 2 \frac{\ln(0.5)}{-\pi} a = 2 \times 0.47a = 0.98a \approx a \quad (3.9)$$

This equation gives us better idea of the parameter a , which represents the half-magnitude gabor filter bandwidth.

All ECoG data sets were visually inspected and re-referenced with respect to the common average to account for any unwanted environmental noise. For these analyses, the time-series ECoG data was converted into the frequency domain using a Gabor filter bank [16]. Spectral amplitudes between 0 and 550 Hz were analyzed on a logarithmic scale. The finger positions from the data glove were converted into velocities.

These frequency responses and velocities were then used as an input to machine learning algorithms described below. Inherent in this is the estimation of the lag between the ECoG signal and the actual finger movement. As part of the modeling process, the value of this variable which resulted in the best decoding accuracy was chosen for further analysis. Average time lags were then used to align the ECoG signal to the finger movement signal. Those features optimized for predicting individual finger movement were then reviewed in light of anatomic location and spectral association in each subject.

3.2 Dimensionality Reduction

Due to the high dimensionality of the spectral data ($\#channels(N) \times \#frequencies(F)$), it is important to reduce the dimensions in order to build a more conducive machine learning algorithm. Principle component analysis, or PCA, is among the most popular dimensionality reduction algorithm. PCA projects the original high-dimensional feature space into a much lower *principle subspace*, such that the variance of low-dimensional data is maximized. Each principle component is an orthogonal vector. Figure 3.1 shows an example of using PCA for 2-D data. The bold red axis is the first principle component, and the bold black axis is the second principle component. In order to reduce the dimensionality of the 2-D data, we can project the data points to the bold red axis.

Let us formulate the definition of PCA method. Given a set of data $X = \{x_i \in R^d\}$ in space D , where $i = 1, \dots, N$, d is the dimensionality of the original feature space of D . Our goal is to find a low-dimensional embedding $Y = \{y_i \in R^m\}$ in space M , where m is the dimensionality of the projected feature space M , and y_i is the projected data point in space M . Let us find the first component U_1 . The projection on the first component U_1 can be written as a linear combination of X , such that:

$$U_1 = \mathbf{w}^\top X \tag{3.10}$$

$$var(U_1) = var(\mathbf{w}^\top X) = \mathbf{w}^\top X (\mathbf{w}^\top X)^\top = \mathbf{w}^\top S \mathbf{w} \tag{3.11}$$

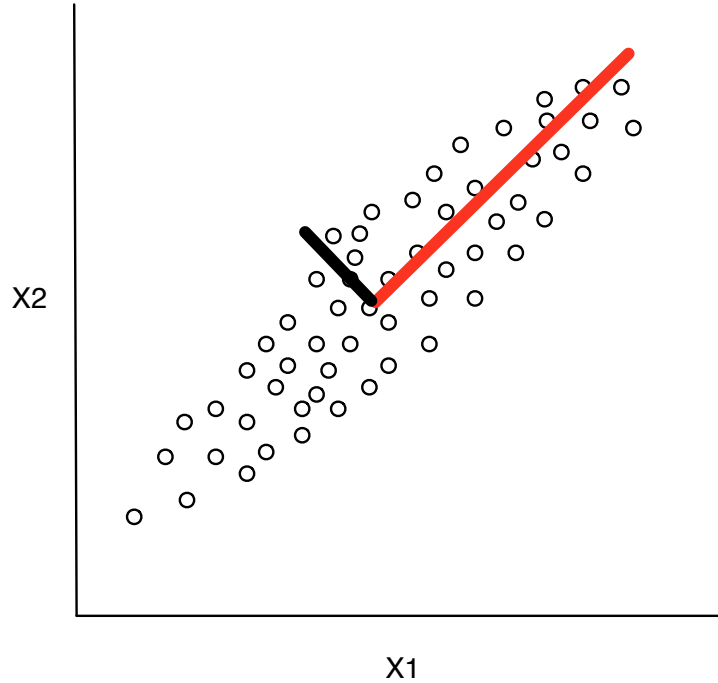


Figure 3.1: Example of PCA for 2-D data.

where, $S = XX^T$ is a $d \times d$ covariance matrix of X . Since we want the principle component has the largest variance, we formulate the following optimization problem:

$$\text{Maximize: } \mathbf{w}^T S \mathbf{w} \quad (3.12)$$

$$\text{Subject to: } \mathbf{w}^T \mathbf{w} = 1 \quad (3.13)$$

Using the Lagrange multiplier, we get as follow:

$$L(\mathbf{w}, \lambda_1) = \mathbf{w}^T S \mathbf{w} - \lambda_1 (\mathbf{w}^T \mathbf{w} - 1) \quad (3.14)$$

Let $\frac{\partial L(\mathbf{w}, \lambda_1)}{\partial \mathbf{w}} = 0$:

$$S \mathbf{w} = \lambda_1 \mathbf{w} \quad (3.15)$$

Multiplying both side by \mathbf{w}^T gives us

$$\mathbf{w}^T S \mathbf{w} = \lambda_1 \mathbf{w}^T \mathbf{w} = \lambda_1 \quad (3.16)$$

We notice that to have the maximum variance, the eigenvector is actually U_1 , and the eigenvalue is the maximum variance $var(U_1)$. In order to get the top m principle components, we can take the top m eigenvectors and their corresponding eigenvalues.

In the real-time decoding task, we use PCA to reduce the input data. However, in the weight analysis, we preserve all the features without reducing the dimensionality because we want to study the effect of using all the features.

3.3 Electrode Co-Registration

Radiographs were used to identify the stereotactic coordinates of each grid electrode [11], and cortical areas were defined the GetLOC package for ECoG electrode localization [18]. Stereotactically defined electrodes were mapped to the standardized brain model. The experimental results were then collated with these anatomical mapping data.

Chapter 4

Machine Learning Algorithms

In this section, we describe the machine learning algorithms used for the finger movement decoding tasks. We focus on three different settings: 1. binary classification, 2. multitask classification, 3. regularized linear regression for kinematic decoding. All the data is split into a training and a testing dataset. We chose our parameters based on a validation dataset split from the training dataset.

4.1 Notations

In this work, we treat the detecting finger movement as a classification problem: given an input data D , predicts the class. The input data is a time-series (\vec{x}_i, y_i) , $i = 1, 2, \dots, T$, where T is the number of data point in the time-series. $\vec{x}_i \in \mathcal{R}^d$ is the frequency spectrum at time i , and d is the total number of frequency features; y_i is a binary label, where $+1$ stands for finger movement, and -1 for no movement at time i . The general goal is to find a classifier Φ , such that $\Phi : \Phi(\vec{x}_i) \rightarrow y_i$. As for kinematic decoding, there is a slight change. Instead of a binary label, y_i is the *position* of the finger at time i , and is normalized between 0 and 1. The goal here is to find a linear mapping $\Phi : \Phi(\vec{x}_i) \rightarrow y_i$. The whole data is split into training, validation and testing dataset, and the parameters of the models are chosen using the validation dataset.

4.2 ℓ_1 -Regularized Logistic Regression

4.2.1 Logistic Regression for Binary Classification

Logistic regression is a generalized linear model for binomial regression. It predicts the posterior probability of a class C by fitting the data to a logistic sigmoid function such that $p(C|\vec{x}) = y(x) = \sigma(w^T \vec{x}_i)$. Here $\sigma(\cdot)$ is the logistic function (or sigmoid function), and is defined by

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (4.1)$$

which is plotted in Figure 4.1. From the figure, we can see the input of the logistic function can be from $-\infty$ to ∞ , and the output is always between 0 and 1. An important property of the logistic function is the symmetry property

$$\sigma(-z) = 1 - \sigma(z) \quad (4.2)$$

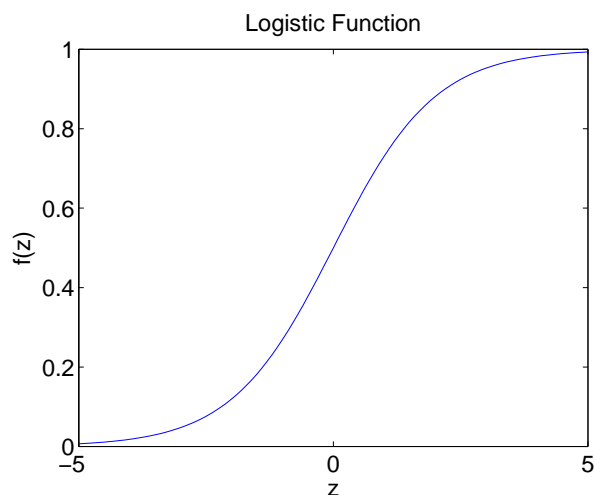


Figure 4.1: Logistic function. With input value z on the x-axis, and output value $f(z)$ on the y-axis.

Usually the input z is defined as a linear combination such that $z = \mathbf{w}^T \vec{x}_i + b$, where $\mathbf{w} \in \mathcal{R}^d$ is the regression coefficients, b is the bias term of the linear model. Given $\vec{x}_i \in \mathcal{R}^d$ at time i and its target label $y_i \in \{+1, -1\}$, we model the posterior

probability of the class C given \vec{x}_i and \mathbf{w} as

$$p(y_i = 1|\vec{x}_i; w, b) = \frac{1}{1 + \exp(-(\mathbf{w}^T \vec{x}_i + b))}, i = 1, 2, \dots, N \quad (4.3)$$

$$\begin{aligned} p(y_i = -1|\vec{x}_i; w, b) &= (1 - p(y_i = 1|\vec{x}_i; w, b)) \\ &= \frac{1}{1 + \exp(\mathbf{w}^T \vec{x}_i + b)}, i = 1, 2, \dots, N \end{aligned} \quad (4.4)$$

Provided the property in Equation 4.2, Equation 4.3 and Equation 4.4 can be written as

$$p(y_i|\vec{x}_i; w, b) = \frac{1}{1 + \exp(-y_i(\mathbf{w}^T \vec{x}_i + b))}, i = 1, 2, \dots, N \quad (4.5)$$

where $\mathbf{w} \in \mathcal{R}^d$ and b is the coefficients and bias of the linear model, $\vec{x}_i \in \mathcal{R}^d$ is the data point at time t , and y_i is the label (+1 stands for movement, and -1 stands for no movement). To solve the problem in Equation 4.5, our goal is to minimize $\prod_1^N p(y_i = 1|\vec{x}_i; \mathbf{w}, b)$, which is equivalent to the following optimization problem

$$\min \sum_{i=1}^N \log(1 + \exp(-y_i(\mathbf{w}^T \vec{x}_i + b))) \quad (4.6)$$

4.2.2 Overfitting and Regularization

Like all linear models, the potential hazard of overfitting exists in logistic regression. To solve this overfitting issues, regularization technique is often used. It adds explicitly a penalization term to the objective function in order to make a tradeoff between the reducing complexity of model and the degree of fitting the training data.

In general, the regularized objective function can be written as

$$Loss + \lambda \|\mathbf{w}\|_q \quad (4.7)$$

where q is the norm of \mathbf{w} . $q = 1$ stands for ℓ_1 regularization (or Lasso), and $q = 2$ stands for ℓ_2 regularization (or Ridge regression).

Given a set of training sample points (\vec{x}_i, y_i) from the ECoG signals, our goal is to solve an optimization problem of the form

$$\min \sum_{i=1}^N \log(1 + \exp(-y_i(\mathbf{w}^T \vec{x}_i + b))) + \lambda \|\mathbf{w}\|_q \quad (4.8)$$

where λ is the regularization constant, and $\|\mathbf{w}\|_q$ is the ℓ_q norm. λ controls the model parameters such that the resulting model makes tradeoff between degree of overfitting and reducing complexity of the model. Larger value of λ gives more penalty on complexed models and hence yields more generalization of the model and vice versa.

4.2.3 ℓ -1 norm vs ℓ -2 norm

ℓ -1 norm and ℓ -2 norm are mostly used in regularization. The pros and cons of the two norm is explained in detail in [20]. In general, ℓ -1 norm is particularly useful in feature selection. Given a dataset with high dimensionality D , there may only be a few subset of feature with r -dimension that is most informative in classification tasks. In this scenario, by tuning the regularization constant λ , we will have sparse representation of the data. However, the difficulty of using ℓ -1 norm lies in the fact that it is hard to optimize at zero point. There are several solver dealing with ℓ -1 regularized optimization problems [15]. On the other hand, ℓ -2 norm is much faster to optimize, since it is differentiable everywhere. However, the drawback of using ℓ -2 norm in feature selection is that ℓ -2 norm always puts more penalty on the features with larger weights, and push all the feature to small values.

4.3 Support Vector Machines

Support vector machines [6], or SVM, is currently the state-of-art classification algorithm in machine learning research. The idea of SVM is to construct a hyperplane in the high-dimensional space for classification. Intuitively, a good hyperplane is the one that separates two classes by maximum margin. The idea of maximum margin leads to better generalization ability.

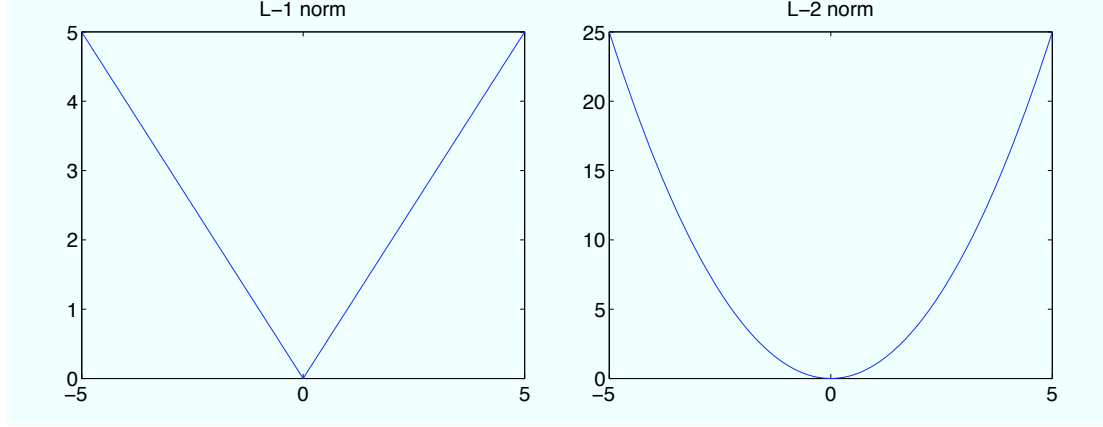


Figure 4.2: Different norms of regularization term. ℓ -1 norm treats each feature equally the same; while ℓ -2 norm gives more penalty on the features with larger weights.

Formally, given a set of data: $(\vec{x}_i, y_i), (i = 1, 2, \dots, N)$, where $\vec{x}_i \in \mathcal{R}_f$. Let $y_i \in (+1, -1)$. Consider all positive data points (\vec{x}_i^+, y_i^+) and all negative data points (\vec{x}_i^-, y_i^-) , we define two parallel boundaries H^+ and H^- :

$$\begin{aligned} H^+ &: \mathbf{w}^\top \vec{x}_i^+ + b = 1 \\ H^- &: \mathbf{w}^\top \vec{x}_i^- + b = -1 \end{aligned} \quad (4.9)$$

such that

$$\begin{aligned} \mathbf{w}^\top \vec{x}_i + b &\geq 1, \text{ if } y_i = 1 \\ \mathbf{w}^\top \vec{x}_i + b &\leq -1, \text{ if } y_i = -1 \end{aligned} \quad (4.10)$$

Thus, the margin is computed as:

$$margin = \frac{2}{\|\mathbf{w}\|} \quad (4.11)$$

However, this is only an idea scenario in linear separable cases. When dealing with linear non-separable cases, we need to further loose the constraint in Equation 4.10.

Thus, slack variable ξ is introduced to allow violation of the constraint, i.e.

$$\begin{aligned} \mathbf{w}^\top \vec{x}_i + b &\geq 1 - \xi_i, \text{ if } y_i = 1 \\ \mathbf{w}^\top \vec{x}_i + b &\leq -1 + \xi_i, \text{ if } y_i = -1 \end{aligned} \quad (4.12)$$

where, $\forall i, \xi_i \geq 0$.

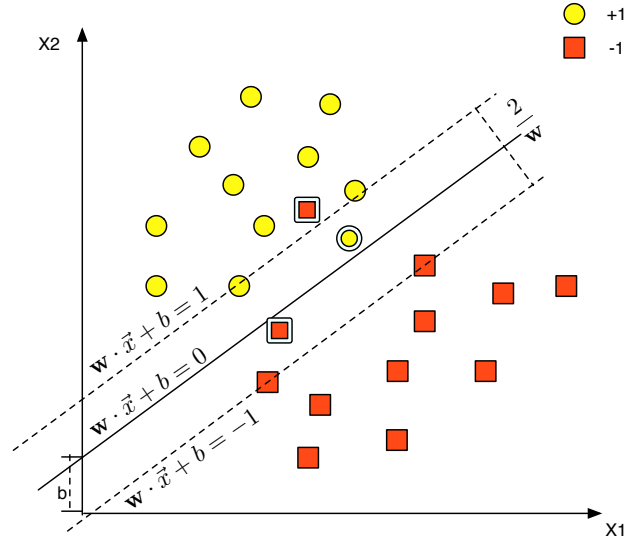


Figure 4.3: Example of SVM in 2D space. The circles are the positive examples, and squares are the negative examples. The examples with double stroke lines around are the ones that violate the strict constraint of linear SVM.

Figure 4.3 shows an example in 2-D space. Positive examples are marked as circles, and negative examples are marked as squares. The concrete black line is the hyperplane (a line here in 2-D space), and the two dotted lines are where the support vectors lie on. The distance between the two dotted lines are the margin we want to maximize. The examples with double stroke lines are counted in the slack variables, and thus alleviate the strict constraint in Equation 4.10.

We can formally develop the problem in Equation 4.12 as the following optimization problem:

$$\text{Minimize: } f(\mathbf{w}) = \lambda \|\mathbf{w}\|_q + \sum_{i=1}^N \xi_i \quad (4.13)$$

$$\text{Subject to: } \forall i, y_i(\mathbf{w}^\top \vec{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \quad (4.14)$$

where, $\xi_i = [1 - y_i(\mathbf{w}^\top \vec{x}_i + b)]_+$, and $[a]_+ = \max(a, 0)$ is the hinge loss function. λ is the regularization constant that controls the complexity of the model and degree of fitting training data. q is the norm of the \mathbf{w} . We use a conjugate gradient descend algorithm by Carl Edward Rasmussen to solve the optimization problems. ¹

Here we discuss $q = 2$ case a little bit more. For the purpose of illustration, let us rewrite Equation 4.13:

$$\text{Minimize: } \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^N \xi_i \quad (4.15)$$

$$\text{Subject to: } \forall i, y_i(\mathbf{w}^\top \vec{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \quad (4.16)$$

The decision boundary is given as $\mathbf{w}^\top \vec{x} + b = 0$. We call this optimization problem as the primal problem. We can solve it using Lagrange multipliers as follows:

$$L_P = \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i(\mathbf{w}^\top \vec{x}_i + b) - 1 + \xi_i] - \sum_{i=1}^N \mu_i \xi_i \quad (4.17)$$

¹<http://www.gatsby.ucl.ac.uk/~edward/code/minimize/minimize.m>

Here, α_i and μ_i are the Lagrange multipliers. The Kuhn-Tucker conditions for optimality are given as:

$$\frac{\partial L_P}{\partial w_j} = w_j - \sum_{i=1}^N y_i \alpha_i x_{ij} = 0 \quad (4.18)$$

$$\frac{\partial L_P}{\partial b} = - \sum_{i=1}^N y_i \alpha_i = 0 \quad (4.19)$$

$$\frac{\partial L_P}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \quad (4.20)$$

$$y_i(\mathbf{w}^\top \vec{x}_i + b) - 1 + \xi_i \geq 0 \quad (4.21)$$

$$\xi_i \geq 0 \quad (4.22)$$

$$\alpha_i \geq 0 \quad (4.23)$$

$$\mu_i \geq 0 \quad (4.24)$$

$$\alpha_i [y_i(\mathbf{w}^\top \vec{x}_i + b) - 1 + \xi_i] = 0 \quad (4.25)$$

$$\mu_i \xi_i = 0 \quad (4.26)$$

where x_{ij} is the j^{th} feature of the i^{th} data. Thus, the duality of Equation 4.15 is given as:

$$\text{Maximize: } L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \langle \vec{x}_i \cdot \vec{x}_j \rangle \quad (4.27)$$

$$\text{Subject to: } \forall i, \sum_{i=1}^N y_i \alpha_i = 0, 0 \leq \alpha_i \leq C \quad (4.28)$$

The decision boundary, similar to the one in primal problem, is given as

$$\sum_{i=1}^N y_i \alpha_i \langle \vec{x}_i \cdot \vec{x} \rangle + b = 0 \quad (4.29)$$

4.4 Kernel SVM

SVM is a powerful classifier, with one big caveat: it cannot handle non linear-separable classification very well. Kernel method is one of the most powerful tool

in machine learning, and in particular, a powerful compliment for support vector machines. The general approach of kernel method is to project the data in the original feature space into a high dimensional feature space via a linear or non-linear mapping ϕ , i.e. $X \in R^d \rightarrow F \in R^m$, where d is the dimensionality of the original feature space, and m is the dimensionality of the projected feature space. When $m = +\infty$, the projected space is called Hilbert space.

After the mapping, the SVM can be written as the following equation:

$$\text{Minimize: } \frac{1}{2} \mathbf{w}^\top \mathbf{w} + C \sum_{i=1}^N \xi_i \quad (4.30)$$

$$\text{Subject to: } \forall i, y_i(\mathbf{w}^\top \phi(\vec{x}_i) + b) \geq 1 - \xi_i, \xi_i \geq 0 \quad (4.31)$$

The corresponding duality problem is:

$$\text{Maximize: } L_D = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \langle \phi(\vec{x}_i) \cdot \phi(\vec{x}_j) \rangle \quad (4.32)$$

$$\text{Subject to: } \forall i, \sum_{i=1}^N y_i \alpha_i = 0, 0 \leq \alpha_i \leq C \quad (4.33)$$

For any test data \vec{x} , the decision is made by the boundary

$$\sum_{i=1}^N y_i \alpha_i \langle \phi(\vec{x}_i) \cdot \phi(\vec{x}) \rangle + b \quad (4.34)$$

Note that we do have a potential problem in using kernel methods: the explicit mapping ϕ can be computationally heavy. **Kernel tricks** allow us to compute the inner product of $\langle \phi(\vec{x}) \cdot \phi(\vec{z}) \rangle$ without actually knowing what ϕ really is. Here we define kernel function K as: $K(\vec{x}, \vec{z}) = \langle \phi(\vec{x}) \cdot \phi(\vec{z}) \rangle$.

In the previous section, we already use a simplest kernel function - linear kernel function $K(\vec{x}_i, \vec{x}) = \langle \vec{x}_i \cdot \vec{x} \rangle$. Here we list other commonly used kernel functions:

Polynomial Kernel: $K(\vec{x}, \vec{z}) = (\langle \vec{x}, \vec{z} \rangle + \theta)^d$, where d is the degree of the kernel function, and $\theta > 0$.

Gaussian RBF Kernel: $K(\vec{x}, \vec{z}) = \exp(-\gamma \|\vec{x} - \vec{z}\|^2)$, where γ is the parameter of the Gaussian kernel.

Sigmoid Kernel: $K(\vec{x}, \vec{z}) = \tanh(\gamma \langle \vec{x}, \vec{z} \rangle + \theta)$, where γ is the parameter of the Gaussian kernel, and $\theta > 0$.

4.5 Multiclass Classification

Besides binary classification tasks, we also need the setup for K -class classification scenario. We adopt the Crammer and Singer multi-class adaptation of support vector machines (MCSVM) [7]. For each class $k \in \{1, \dots, K\}$, we learn class-specific parameters \mathbf{w}_k, b_k . The loss only focuses on pairwise comparisons between the different classes and ensures that $\mathbf{w}_k^\top \vec{x}_i + b_k \geq \mathbf{w}_r^\top \vec{x}_i + b_r + 1$ if $y_i = k$ for any $r \neq k$. For completeness, we re-state the optimization problem:

$$\min_{(\mathbf{w}_1, b_1), \dots, (\mathbf{w}_K, b_K)} \sum_{i=1}^N \sum_{r \neq y_i} \max(1 + \mathbf{w}_r^\top \vec{x}_i + b_r - (\mathbf{w}_{y_i}^\top \vec{x}_i + b_{y_i}), 0) + \lambda \sum_{k=1}^K \|\mathbf{w}_k\|_q. \quad (4.35)$$

Similar to the scenario of binary classification, the constant $\lambda \geq 0$ regulates the trade-off between complexity and sparseness. In multiclass classification, kernel method can also be applied.

4.6 LSR with Min-Max Constraint

In the movement traces prediction setting, we want to predict the position of each finger y_t (which is normalized between 0 and 1). The goal is to predict the position y_t of fingers, given the feature vector \mathbf{x}_t . For this purpose, we use a modified regularized least square regression (LSR). The predictors learn parameter $(\mathbf{w}, b) \in \mathcal{R}^d \times \mathcal{R}$ by

solving the following optimization problem

$$\min_{(\mathbf{w}, b)} \sum_{t=1}^T (\min(\max(\mathbf{w}^\top \mathbf{x}_t + b, 0), 1) - y_t)^2 + \lambda \|\mathbf{w}\|_q. \quad (4.36)$$

The reason why we are thresholding the term $\mathbf{w}^\top \mathbf{x}_t + b$ between 0 and 1 is because given the output of the predictor is above 1 or below 0, we do not want to put extra penalty on the square loss. Here, the constant $\lambda \geq 0$ regulates the trade-off between complexity and sparseness.

4.7 Multitask Learning

4.7.1 Introduction to Multitask Learning

In many practical situations, we need to infer multiple models from the data. For example, in computer vision research, we may be interested in learning multiple classifiers for detecting faces, flowers, and cars; In data mining research, we may need to learn patterns in the data from multiple domains; in speech recognition, we may need to learn language models for different foreign accents. The basic intuition behind multitask learning is that if there exists similarity across different tasks, it would be beneficial to learn multiple models jointly.

In [3], Caruana first introduced multitask learning into neural network. Later, Evgeniou et al introduced regularized multitask learning method [9], where they extend SVMs to multitask learning setting. In [5], multitask learning is applied to boosting with application to web search ranking. In [21], multitask learning is applied to the large margin nearest neighbor algorithm [29]. In all these works, the authors show multitask learning perform better than their single-task learning counterpart.

In our work, the traditional way to detect finger movement is to build one classifier for each finger. However, it seems reasonable to assume that there exists brain activity that is predictive of general finger movement, independent of the exact finger. Although each individual finger movement is associated with its specific anatomical and spectral features, there are certain features which are associated with the general

cortical processing of finger movements. This is analogous to the notion of language processing and articulation in cortical areas. Functional magnetic resonance imaging (fMRI) studies have shown that although speech is represented in general cortical areas, individual features specific to different kinds of words can be found [14, 22]. We adopt the MTL adaptation for SVMs of [9], and an analogous framework for logistic regression, which leverages the commonalities across learning tasks by modeling them explicitly with an additional shared weight vector. We will explain the implementation of the two MTL adaptation in the following sections. In Figure 4.4 and Figure 4.5, we show the general framework of single-task learning and multi-task learning.

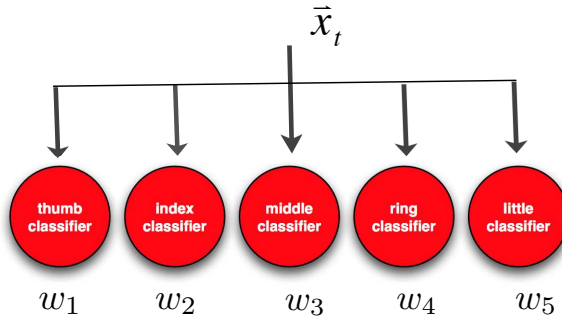


Figure 4.4: Single-task learning for detecting finger movements.

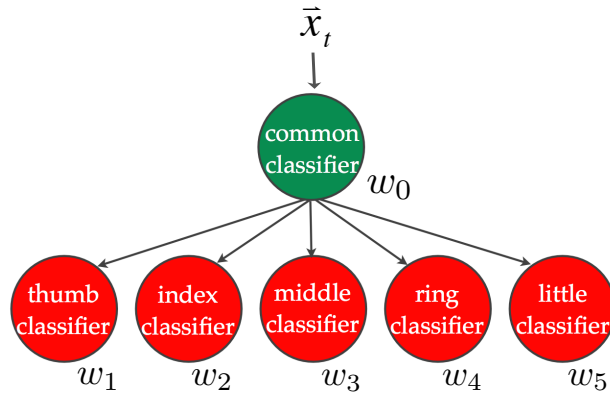


Figure 4.5: Multi-task learning for detecting finger movements. The green circle is the shared representation \mathbf{w}_0 of all fingers' movements, and the red circles $\mathbf{w}_i, i = 1, 2, \dots, k$ are the specific representation of finger i . The prediction at time t for finger k is defined as $y_t = (\mathbf{w}_0 + \mathbf{w}_k)^\top \bar{x}_t$.

4.7.2 SVM-MTL and LR-MTL

In this section, we will demonstrate the SVM-MTL algorithm. Given K classification tasks, let us define \mathbf{w}_0 as the shared weight of the K classifier. $\mathbf{w}_{\mathbf{k}>0}$ is the specific weight of classifier k . y_{ki} is the label of the task k , where $k = 1, 2, \dots, K$, $i = 1, 2, \dots, N$. Based on Equation 4.13, we adapt multitask learning and rewrite the SVM-MTL setting as the following:

$$\begin{aligned} \text{Minimize: } & \lambda_0 \|\mathbf{w}_0\|_q + \sum_{k=1}^K \lambda_k \|\mathbf{w}_{\mathbf{k}>0}\|_q + \sum_{k=1}^K \sum_{i=1}^N \xi_{ki} \\ \text{Subject to: } & \forall k, i, y_{ki}((\mathbf{w}_0 + \mathbf{w}_{\mathbf{k}})^T \vec{x}_i + b_0 + b_k) \geq 1 - \xi_{ki}, \xi_{ki} \geq 0 \end{aligned} \quad (4.37)$$

Similarly, we have the MTL adaptation for logistic regression.

$$\text{Minimize: } \sum_{i=1}^N \log(1 + \exp(-y_{ki}((\mathbf{w}_0 + \mathbf{w}_{\mathbf{k}})^T \vec{x}_i + b_0 + b_k))) + \lambda_0 \|\mathbf{w}_0\|_q + \sum_{k=1}^K \lambda_k \|\mathbf{w}_{\mathbf{k}>0}\|_q \quad (4.38)$$

In both Equation 4.37 and Equation 4.38, λ_0 is the regularization constant for shared weight, and λ_k is the regularization constant for classifier k . Note that the parameter λ_0 regulates how much of the learning is shared. If $\lambda_0 \rightarrow +\infty$, then $\mathbf{w}_0 = \mathbf{0}$ and we reduce our setting to the original binary classification mentioned above. On the other hand, setting $\lambda_0 = 0$ and $\lambda_{k>0} \gg 0$ will result in weight vectors $\mathbf{w}_{\mathbf{k}>0} = \mathbf{0}$. As a result, one would learn only a single classifier with weight vector \mathbf{w}_0 for generic finger movement.

Chapter 5

Results

5.1 Evaluation Metric

In this work, we are using AUC as our evaluation metric. This is the area under the receiver operating characteristic curve, which is a plot of sensitivity (true positive rate) with respect to $1 - \text{specificity}$ (false positive rate) (in Figure 5.1). This is widely used for binary classification since it allows user to specify a discrimination threshold for decision making. It is a better evaluation metric than accuracy when the number of the positive examples is much less than that of the negative examples. Note that the AUC of a random classifier (red dotted line) is 0.5, the AUC of a perfect classifier (green solid line) is 1, and a *good* classifier (blue solid line) should have an AUC between 0.5 – 1.

5.2 Time Lag

There is a characteristic delay between brain activity and resulting movement. In the first place, we studied the effect of this delay between the ECoG signal and the actual movements. A set of decoding accuracy is obtained by shifting the feature dataset \vec{x}_t and the target label y_t by a presumed time-lag (i.e. we evaluated the performance of decoder: $h : h(\vec{x}_t) = y_{t+\delta_T}$, by increasing the value of δ_T .)

In Figure 5.2, we show the decoding accuracy as a function of time-lag for individual finger movement. The bolded line is the average decoding performance of individual

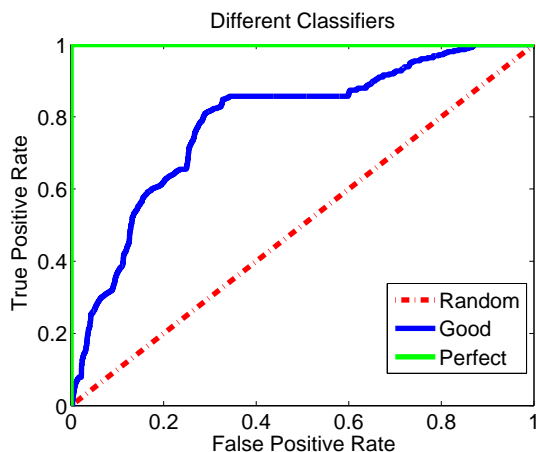


Figure 5.1: Receiver Operating Characteristics. X-axis is the false positive rate, and y-axis is the true positive rate.

fingers with respect to different time-lags. Time-lag is selected from 0 to 800 milliseconds and the best decoding time-lag is selected as the value of δ_T that leads to the best decoding performance. In this case, the average time lag for the ipsilateral finger movement is around 158 ms. This is in accordance with previous studies by our group which show that the ipsilateral cortical activity precedes actual movement on an average by 160 ms [30]. We have fixed the decoding time lag for the ipsilateral finger movement decoding throughout the paper. The longer time-lag also confirms that we are using the signals from the ipsilateral cortex directly, rather than using the signals bypassed from the contralateral cortex.

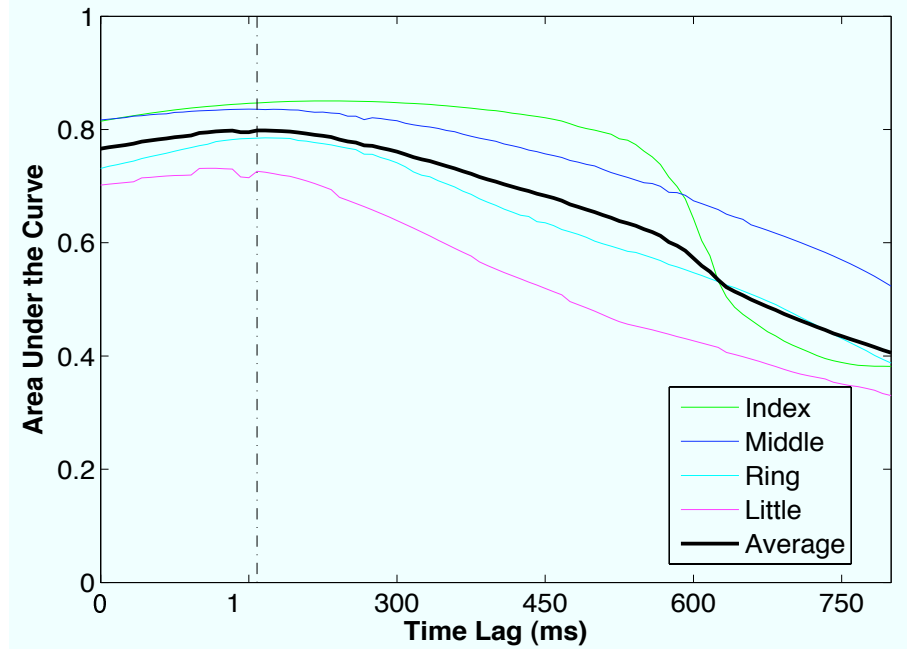


Figure 5.2: Different time-lag and its corresponding decoding performance. X-axis is the value of time-lag (seconds), and y-axis is the AUC.

5.3 Detecting Finger Movement

We characterize the movement detection task as a binary classification. We first set a threshold $thresh$, and label the targets y_t as 1 if the velocity at time t $v_t \geq thresh$, and -1 otherwise. Then, we use ℓ_1 -regularized logistic regression for the binary classification. We use receiver operating characteristic (ROC) curve to evaluate the performance of the binary classification. ROC curve is widely used in signal estimation and detection theory, and is a graphical plot of true positive rate versus the false positive rate. ROC analysis allows user to pick the optimal discrimination threshold for the binary classifier. We pick regularizer λ from validation dataset. Figure 5.1 shows the result of ROC curve for three subjects. This demonstrates that ℓ_1 -regularized logistic regression is a powerful tool in detecting finger movement.

5.4 Learning Commonality of the Brain Activity

In this section, we present how multitask learning improves the performance of the classifier. Although multitask learning has been employed in the context of brain signal decoding [1], we are the first to decode ECoG signals in humans. We group all the individual finger movement together, such that each task has similarity with others. First of all, we evaluate the performance of single-task learning using SVM. Then, we study the SVM-based multitask learning. As we show in Equation 4.37 and in Equation 4.38, we make trade-off between modeling joint component and modeling class-specific components by adjusting parameters λ_0 and $\lambda_{k>0}$. We search a number of regularization constant ($\lambda_0, \lambda_{k>0}$), and pick up the parameters that lead to highest average AUC for all tasks. We compare the ℓ_1/ℓ_2 -regularized logistic regression-based multitask learning (LR-MTL) with ℓ_1/ℓ_2 -regularized SVM-based multitask learning (SVM-MTL). Table 5.1, Table 5.2, and Table 5.3 shows the multitask learning versus single-task learning with different classifiers (ℓ_1/ℓ_2 -regularized logistic regression-based multitask learning, and ℓ_1/ℓ_2 -regularized SVM-based multitask learning). In most cases, multitask learning helps classification performance. This confirms our assumption that there exists brain activity that controls the finger movement, irrespective of any particular finger. By carefully searching the best parameters that regulates the trade-off between learning commonality among all finger

movement and specificity of exact finger movement, the classification algorithm can be significantly improved. Again, it illustrates that multitask learning is particularly helpful in learning similar tasks that are controlled by the brain.

Classifier	Thumb	Index	Middle	Ring	Little	Average	Improvement
STL-SVM1	N/A	0.8505	0.8282	0.7581	0.6953	0.7830	16.05%
MTL-SVM1	N/A	0.8322	0.8346	0.8044	0.8002	0.8179	
STL-SVM2	N/A	0.8477	0.8284	0.7479	0.7017	0.7814	25.53%
MTL-SVM2	N/A	0.8494	0.8569	0.8561	0.7865	0.8372	
STL-LR1	N/A	0.8403	0.8276	0.7396	0.6956	0.7758	14.59%
MTL-LR1	N/A	0.8444	0.8325	0.7964	0.7607	0.8085	
STL-LR2	N/A	0.8467	0.8281	0.7403	0.6974	0.7781	1.44%
MTL-LR2	N/A	0.8382	0.8238	0.7472	0.7161	0.7813	

Table 5.1: Multitask learning vs single-task learning for Subject 1.

Classifier	Thumb	Index	Middle	Ring	Little	Average	Improvement
STL-SVM1	0.7710	0.9016	0.9021	0.8888	0.7124	0.8316	9.35%
MTL-SVM1	0.7641	0.8956	0.8998	0.8968	0.8007	0.8514	
STL-SVM2	0.7710	0.9061	0.9021	0.8888	0.7124	0.8316	5.60%
MTL-SVM2	0.7845	0.8948	0.8990	0.8894	0.7586	0.8453	
STL-LR1	0.7537	0.9060	0.8542	0.8809	0.5795	0.7949	8.84%
MTL-LR1	0.6619	0.8931	0.8893	0.8906	0.7301	0.8130	
STL-LR2	0.7685	0.9060	0.8560	0.8862	0.5968	0.8027	5.71%
MTL-LR2	0.6644	0.8932	0.8896	0.8907	0.7319	0.8140	

Table 5.2: Multitask learning vs single-task learning for Subject 2.

5.5 Feature Analysis

An important part of decoding finger movements from cortical activity is to map the features back to cortical domain. Physiologically, it is important to understand the features which contribute most to the decoding algorithms i.e. the features with the highest weights. As shown in Table 5.4 below, the decoding accuracy, indicated by AUC, does not change much as we increase the number of features used for classification. This signifies that from the large feature set used for decoding, a few

Classifier	Thumb	Index	Middle	Ring	Little	Average	Improvement
STL-SVM1	0.8260	0.8044	0.9369	0.6398	0.7635	0.7941	6.68%
MTL-SVM1	0.7592	0.8303	0.9445	0.6663	0.8391	0.8079	
STL-SVM2	0.7680	0.7454	0.9459	0.7404	0.7705	0.7940	18.67%
MTL-SVM2	0.8611	0.8242	0.9481	0.7479	0.7801	0.8323	
STL-LR1	0.8032	0.7986	0.9443	0.7611	0.7832	0.8181	-4.96%
MTL-LR1	0.7306	0.8424	0.9490	0.6873	0.8360	0.8091	
STL-LR2	0.7872	0.8125	0.9487	0.7570	0.7748	0.8160	-4.88%
MTL-LR2	0.7361	0.8385	0.9459	0.6799	0.8349	0.8071	

Table 5.3: Multitask learning vs single-task learning for Subject 3.

features form the core and are the most important. To visualize these core features, we mapped the top 30 features back to the brain. Figure 5.6 shows the normalized weights from the features used to classify finger movements from non-movements. It is apparent from the figure that the features with the highest weights fall in the DLPFC and premotor areas. This is what we would expect since these two areas are the one’s most involved in the planning of motor movements. As previously reported, the frequency range with the highest weights falls in the lower frequencies in ipsilateral movements [30]. In our case, the frequencies fall in the delta-alpha range. As noted by Tallon-Baudry, attention networks of the brain affect the oscillatory synchrony as low as theta-alpha range frequencies [28].

Another important conclusion inferred from Figure 5.6 is that the potential application of μ -ECoG implanted in humans. (Figure 5.5²). Since the top features are located on a small region of the brain, micro-electrode will replace the large 8×8 , or 8×10 electrode array. This will significantly reduce the potential health hazard in developing brain machine interfaces.

5.6 Kinematic Decoding

It is also of interest to decode the kinematic parameters such as position, velocity, and acceleration of fingers for a robust BCI system. In the contralateral experiment

²The micro-electrode grid image is permitted by courtesy of Professor Justin Williams, University of Wisconsin at Madison.

setups, a few research groups have studied the fidelity of the trajectory of arms and fingers, but so far there is few research studying the kinematic decoding using signals from ipsilateral cortex. Figure 5.7 shows the selected examples of actual finger movement (shown in thin traces) and predicted finger movement (shown in bold traces) for the three subjects. The correlation coefficients are listed in Table 5.4. Though the kinematic decoding has a relatively low fidelity compared to the contralateral experiment setup, it is shown that there is potential to decode the finger movements using ECoG signals from the ipsilateral cortex. The relatively low correlation coefficients lie in two aspects: first, the signals we are using to decode movements is from the ipsilateral cortex; second, the classifiers for each finger are too sensitive, resulting in ambiguity in differentiating individual fingers.

	Subject 1	Subject 2	Subject 3
Thumb	N/A	0.1486	0.0167
Index	0.1154	0.3575	0.2707
Middle	0.0744	0.5502	0.5210
Ring	0.0668	0.3403	0.1838
Little	0.0347	0.0715	0.1841
Average	0.0728	0.2487	0.2353

Table 5.4: Prediction the flexion of finger movements.

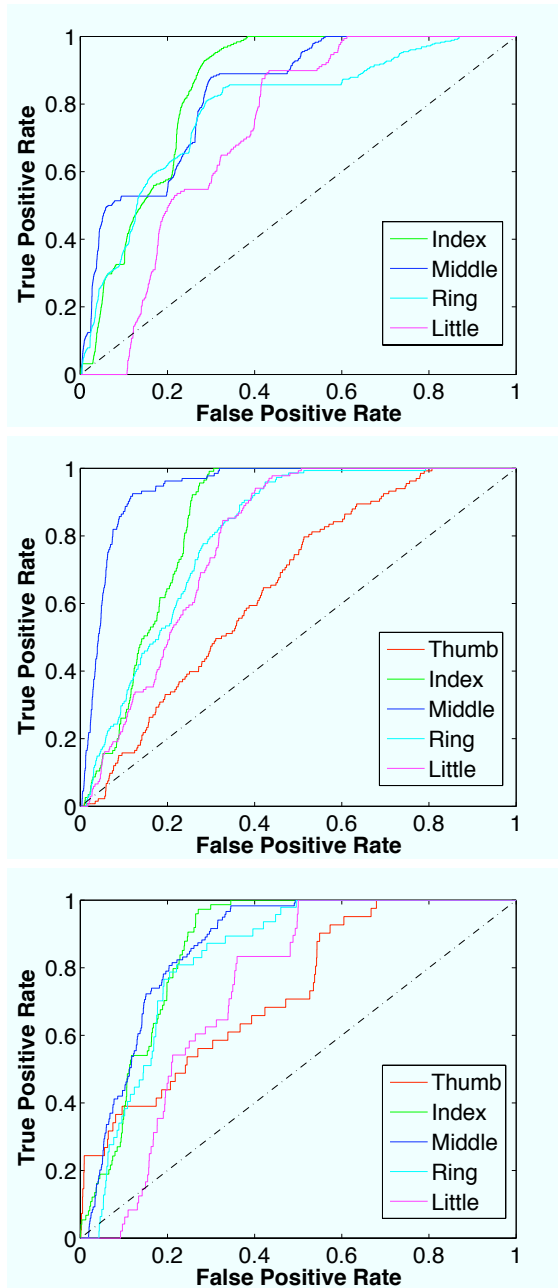


Figure 5.3: ROC curve for the ipsilateral finger movement decoder. Horizontal axis shows the false positive rate, and the vertical axis shows the true positive rate. The dotted line is the accuracy of a random classifier. Classifiers that have higher *area under the ROC curve*, or AUC, indicate better classification performance.

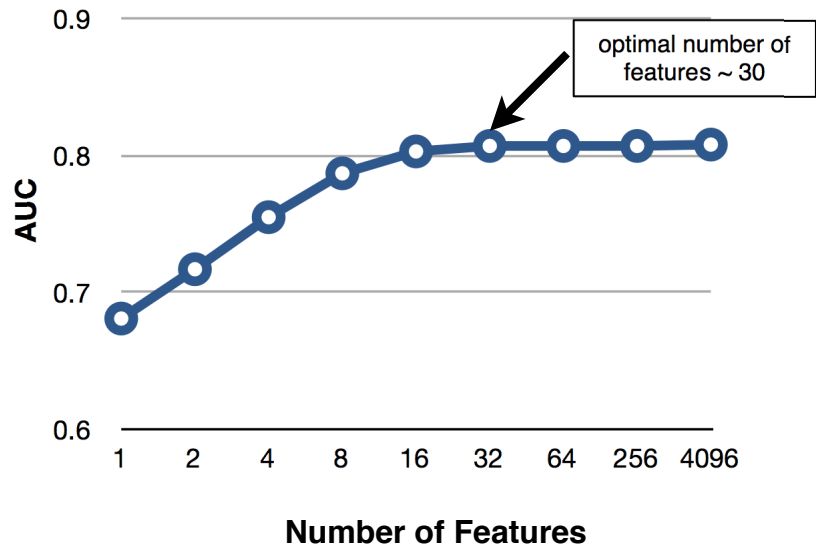


Figure 5.4: The area under the curve (AUC) as a function of the number of features used for classification. Features were selected in decreasing order of their respective absolute weights from logistic regression with ℓ_1 -regularization.

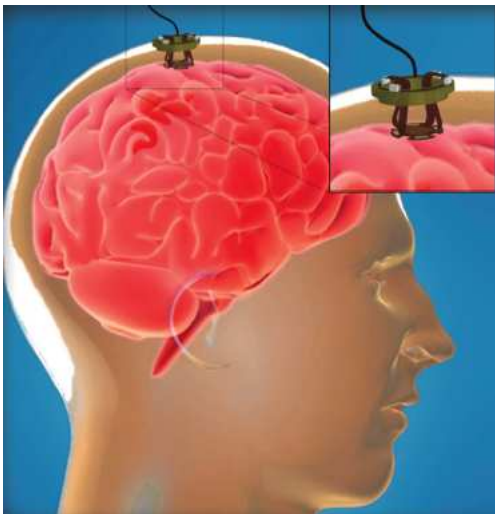


Figure 5.5: μ -ECoG technology: implantation of micro-electrode on human brains.

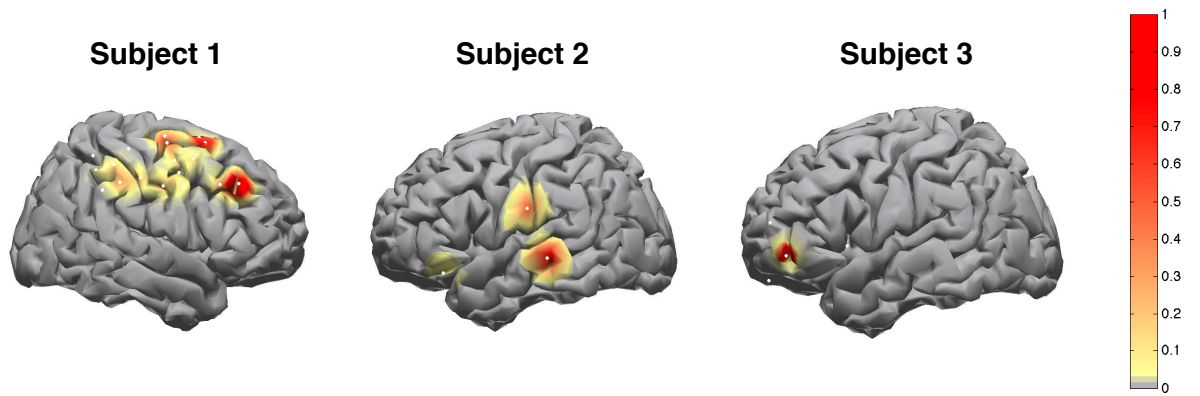


Figure 5.6: Brain map representing the weights of the top 30 features of the three patients.

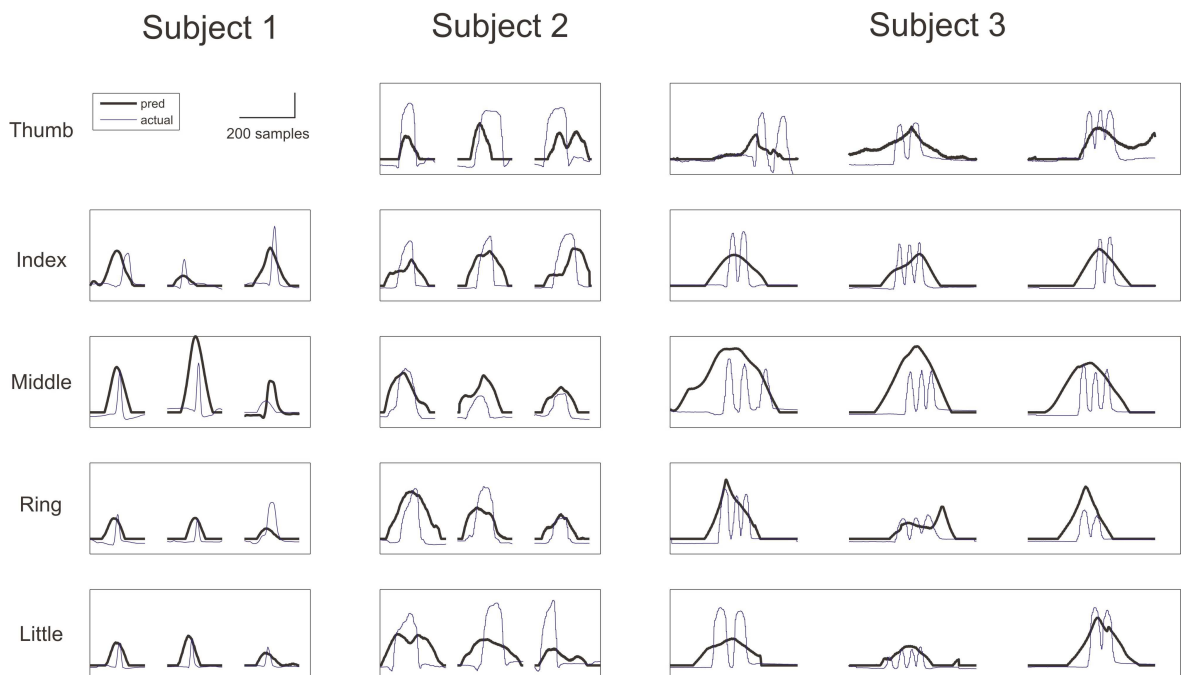


Figure 5.7: Selected prediction of the flexion of finger movements.

Chapter 6

Additional Work

A robust brain computer interface should support many regular movements beyond finger movements. In the real task, a BCI system should be able to interpret the patients' intention to move towards certain directions so as to fetch objects ([25]).

6.1 Experiment Setup for Directional Movement Classification

Figure 6.1 shows the experiment setup. The patient was seated in front of a 17-inch LCD monitor, which gave cues about which direction to move. Once the cue appeared on the monitor, the patient was instructed to move the joystick to one of the eight targets. Once the designated target was reached, the corresponding target turned green and the patient moved the joystick back to the starting point until the next cue came up the monitor. The patient underwent three trials and the total number of movements is 280. However since the brain signal pattern varied from trial to trial, we used the first 225 movements for classification. The ECoG signal was recorded by the implanted electrode grids on the surface of the brain contralateral to the hand controlling the joystick.

During each direction movement task, we further divide it into seven stages as shown in Figure 6.2. State 0 is the resting period between trials. State 1 and State 2 are the baseline of the experiment. State 3 and State 4 are the target encoding states, in which the subject sees the cue and is ready to make the movement. State 5 is the movement state, in which the subject is performing the movement task. State 6 is

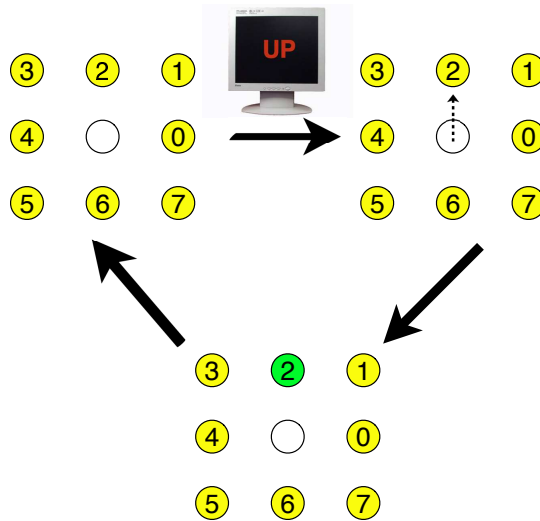


Figure 6.1: The experiment paradigm for directional movement classification.

the reward state, in which the monitor gives a green light if the patient is making the right movement. We have two problems to solve here: First, can we tell the difference between the resting state and the movement state? Second, can we classify the directional movements?

6.2 Classification between Resting and Movement State

One problem of interest is to classify the resting and movement state in the joystick movement task. Here State 0 is the resting period between two consecutive trials, and State 5 is the actual movement period. We use modified ℓ_1 regularized least square regression to classify these two states. Figure 6.3 shows the ROC curve performance on the classification result, and Figure 6.4 shows the predicted joystick movement versus the actual movement. The red line is the prediction on the joystick movement, and the blue bar is the actual joystick movement. The red lines, in most cases, fall in the blue bar, which tells us that we can effectively discriminating between the resting and movement state.

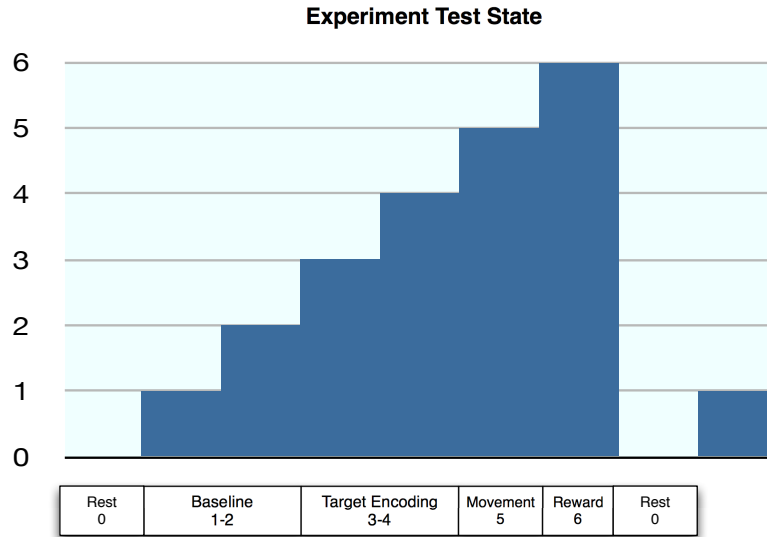


Figure 6.2: The experiment test state.

6.3 Classification between Directions

Another problem of interest is to classify different directions. To be specific, the tasks are classification between moving left and moving right, moving up and moving down, and all of these four directions. In this setting, we perform a feature selection before hand. 22 channels out of 64 and 56 frequency features out of 74 are selected as the most relevant features. We are using State 3 and State 4, which are the target encoding states. Similarly to the scalogram representation in [4], each data point x_i is represented as a vector that contains temporal information, spatial information (channels) and frequency information. Kernel support vector machines are applied here, and we are using a sigmoid kernel. The classification result is shown in Figure 6.5. The blue bar shows the prediction accuracy, and the red line shows the baseline of the prediction by chance. P-value is calculate to test the significance.

Binary Classification of Resting and Movement State

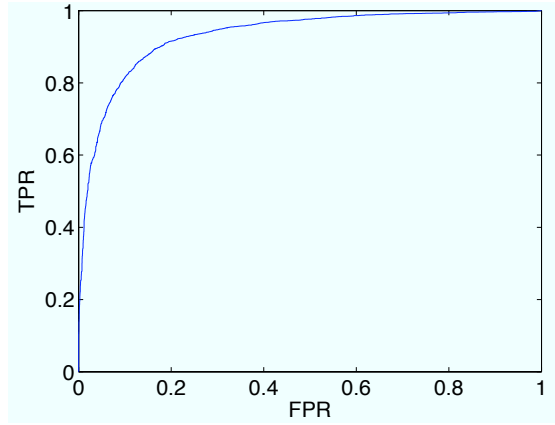


Figure 6.3: The ROC curve of the classification.

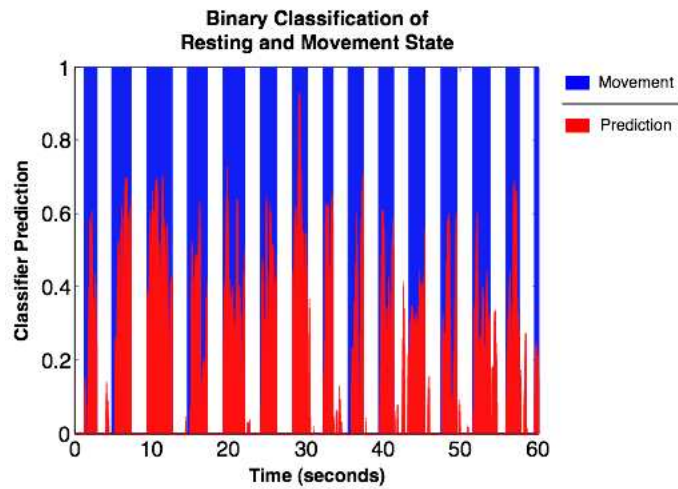


Figure 6.4: Predicted movement versus the actual movement.

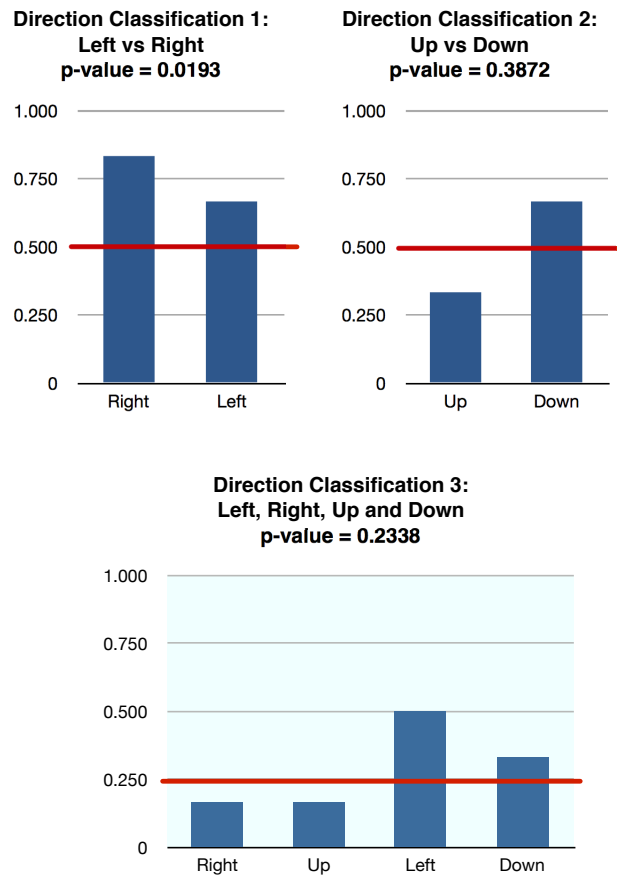


Figure 6.5: Classification between directions.

Chapter 7

Discussion

Few BCI research look into the direct association between ipsilateral cortex and motor-associated tasks. However, ipsilateral cortex proves to be not only participating a planning role in motor-associated tasks, but also an effective predictor for movements. The ipsilateral cortex has a 100 ms longer time-lag than the contralateral cortex in the movement tasks. The work shows promising result on using only ipsilateral cortex to decode finger movements. The hot spot of ipsilateral cortex in motor-associated tasks falls in the prefrontal and pre-motor area of the brain, with the most active frequency bins fall in the low-frequency range. Again, this is different from the previous findings on the contralateral cortex, whose hot spot falls in the motor area, and the most active frequency features fall in higher frequency range.

Using ECoG signals from the ipsilateral cortex, we can efficiently detect finger movements, and further more we can improve the performance of detecting finger movements by introducing the idea of multitask learning in it. We also investigate the large amount of features and extract the sparse representation of features. The result shows that the most contributing spatial features fall in a small region on the large electrode grids. This gives researcher promising idea of using micro-electrode grids to help patients rehabilitate their dysfunction of motor-cortex. Beyond detecting movement, accurate decoding of kinematic parameters is also of huge interest in the sense that these parameters are important for patients to control prosthetic devices with more freedom. In our case, decoding ipsilateral finger position is promising: in a similar setting in the Berlin Brain Computer Interfaces (BBCI) competition, the winning five teams report an average correlation coefficients of 0.46, 0.42, 0.27, 0.10, 0.05. Considering in our case we are decoding from ipsilateral cortex, the result in Table 5.4

shows potential interest for BCI community. The major problem with the kinematic decoding lies in the fact that the classifier is very sensitive, resulting in the unwanted false positives. How to further discriminate among these individual classifier is an open question.

Beyond the finger movement tasks, we also investigate the joystick direction movement tasks. In differentiating between resting and movement state, we see a strong classification result, with very few false positive points. In classifying different direction tasks, the classification results are slightly above chance, and it remains a future research problem.

Machine learning approach is one of the most prominent ways for brain computer interfaces research. However, since most classification algorithms such as support vector machines are data-driven approaches and very subject to scale, feature selection becomes crucial in BCI research. In fMRI research, a number of work on feature selection is reported, such as [24]. In Chapter 3, we use the power spectrum of the ECoG signal, leaving out the phase information in the signal. Recent studies show that the phase information in neurons is very informative in motor-associated tasks. It is worthwhile to regard phase information of the ECoG signal as features. We still have to keep in mind that human brain is an extremely complex mechanism and the patterns are variant to a lot of factors such as time, outside interference, human's emotion, attention and other subjective factors.

Chapter 8

Conclusion

In our work, we have presented a first framework of detecting finger movements from ipsilateral cortex in humans using ECoG signals. This work includes detection of finger movements, a joint learning framework using multitask learning algorithms, and a kinematic decoding framework. The result is inspiring to BCI community, and suggests that there exists potential information in the ipsilateral cortex in humans to decode human movements. This is particularly useful for patients who suffer hemispherical damage due to epilepsy or suffer loss of limbs due to accidents. Moreover, we investigate the directional movement classification tasks and show accurate discrimination between resting and movement state. The future directions of the work are in further discriminating between finger movements, discriminating between different target direction, and real-time decoding of the kinematic parameters. It is still of interest to find the inherent features in the brain that are invariant to variance between the experiment trials and other interference.

References

- [1] M. Alamgir, M. Grosse-Wentrup, and Y. Altun. Multitask learning for brain-computer interfaces. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 9:17–24, 2010.
- [2] E. Buch, C. Weber, L.G. Cohen, C. Braun, M.A. Dimyan, T. Ard, J. Mellinger, A. Caria, S. Soekadar, A. Fourkas, et al. Think to move: a neuromagnetic brain-computer interface (BCI) system for chronic stroke. *Stroke*, 39(3):910, 2008.
- [3] Rich Caruana. Multitask learning. *Machine learning*, 28:41–75, 1997.
- [4] Zenas C Chao, Yasuo Nagasaka, and Naotaka Fujii. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey. *Frontiers in Neuroengineering*, 3(0), 2010.
- [5] Olivier Chapelle, Pannagadatta Shivaswamy, Srinivas Vadrevu, Kilian Weinberger, Ya Zhang, and Belle Tseng. Boosted multi-task learning. *Machine Learning*, pages 1–25, 2010. 10.1007/s10994-010-5231-6.
- [6] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [7] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.
- [8] N.E. Crone, D. Boatman, B. Gordon, and L. Hao. Induced electrocorticographic gamma activity during auditory perception. *Clinical Neurophysiology*, 112(4):565–582, 2001.
- [9] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 109–117, New York, NY, USA, 2004. ACM.
- [10] Theodoros Evgeniou and Massimiliano Pontil. Regularized multi-task learning. In *KDD*, pages 109–117, 2004.
- [11] P.T. Fox, J.S. Perlmutter, and M.E. Raichle. A stereotactic method of anatomical localization for positron emission tomography. *Journal of Computer Assisted Tomography*, 9(1):141, 1985.

- [12] W.J. Freeman, M.D. Holmes, B.C. Burke, and S. Vanhatalo. Spatial spectra of scalp eeg and emg from awake humans. *Clinical Neurophysiology*, 114(6):1053–1068, 2003.
- [13] J.E. Huggins, S.P. Levine, S.L. BeMent, R.K. Kushwaha, L.A. Schuh, E.A. Passaro, M.M. Rohde, D.A. Ross, K.V. Elisevich, and B.J. Smith. Detection of event-related potentials for development of a direct brain interface. *Journal of Clinical Neurophysiology*, 16(5):448, 1999.
- [14] M.A. Just, V.L. Cherkassky, S. Aryal, and T.M. Mitchell. A neurosemantic theory of concrete noun representation based on the underlying brain codes. 2010.
- [15] Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. A method for large-scale l1-regularized logistic regression. In *Proceedings of the 22nd national conference on Artificial intelligence - Volume 1*, pages 565–571. AAAI Press, 2007.
- [16] E.C. Leuthardt, Z. Freudenberg, D. Bundy, and J. Roland. Microscale recording from human motor cortex: implications for minimally invasive electrocorticographic brain-computer interfaces. *Journal of Neurosurgery: Pediatrics*, 27(1), 2009.
- [17] E.C. Leuthardt, G. Schalk, J.R. Wolpaw, J.G. Ojemann, and D.W. Moran. A brain-computer interface using electrocorticographic signals in humans. *Journal of Neural Engineering*, 1:63–71, 2004.
- [18] K.J. Miller, S. Makeig, A.O. Hebb, R.P.N. Rao, M. Dennijs, and J.G. Ojemann. Cortical electrode localization from x-rays and simple mapping for electrocorticographic research: The. *Journal of neuroscience methods*, 162(1-2):303–308, 2007.
- [19] J. R. Movellan. Tutorial on Gabor Filters. *Tutorial paper* <http://mplab.ucsd.edu/tutorials/pdfs/gabor.pdf>, 2008.
- [20] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, ICML '04, pages 78–, New York, NY, USA, 2004. ACM.
- [21] Shubin Parameswaran and Kilian Q. Weinberger. Large Margin Multi-Task Metric Learning. In *NIPS*, December 2010.
- [22] S.E. Petersen, P.T. Fox, M.I. Posner, M.A. Mintum, and M.E. Raichle. Positron emission tomographic studies of the cortical anatomy of single-word processing. *Cognitive psychology: key readings*, page 109, 2004.

- [23] MM Rohde, SL BeMent, JE Huggins, SP Levine, RK Kushwaha, and LA Schuh. Quality estimation of subdurally recorded, event-related potentials based on signal-to-noise ratio. *IEEE Transactions on Biomedical Engineering*, 49(1):31–40, 2002.
- [24] S. Ryali and V. Menon. Feature selection and classification of fmri data using logistic regression with l1 norm regularization. *NeuroImage*, 47:S57, 2009.
- [25] G. Schalk, J. Kubanek, KJ Miller, NR Anderson, EC Leuthardt, JG Ojemann, D. Limbrick, D. Moran, LA Gerhardt, and JR Wolpaw. Decoding two-dimensional movement trajectories using electrocorticographic signals in humans. *Journal of neural engineering*, 4:264–275, 2007.
- [26] G. Schalk, DJ McFarland, T. Hinterberger, N. Birbaumer, and JR Wolpaw. Bci2000: a general-purpose brain-computer interface system. *IEEE Transactions on Biomedical Engineering*, 51(6):1034–1043, 2004.
- [27] R. Srinivasan, PL Nunez, RB Silberstein, E.G. Inc, and OR Eugene. Spatial filtering and neocortical dynamics: estimates of eeg coherence. *IEEE Transactions on Biomedical Engineering*, 45(7):814–826, 1998.
- [28] C. Tallon-Baudry. Oscillatory synchrony and human visual cognition. *Journal of Physiology-Paris*, 97(2-3):355–363, 2003.
- [29] Kilian Q. Weinberger and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. *J. Mach. Learn. Res.*, 10:207–244, June 2009.
- [30] K.J. Wisneski, N. Anderson, G. Schalk, M. Smyth, D. Moran, and E.C. Leuthardt. Unique cortical physiology associated with ipsilateral hand movements and neuroprosthetic implications. *Stroke*, 39(12):3351, 2008.

Vita

Yuzong Liu

Date of Birth September 19, 1987

Place of Birth Nanjing, China

Degrees B.Eng. Electrical and Information Engineering, June 2009
M.S. Computer Science, May 2011

Publications Yuzong Liu, Mohit Sharma, Charles M. Gaona, Jonathan D. Breshears, Jarod Roland, Zachary V. Freudenburg, Kilian Q. Weinberger, and Eric C. Leuthardt. (2010). Decoding Ipsilateral Finger Movements from ECoG Signals in Humans, *Advances in Neural Information Processing Systems* **23**: 1468-1476.

May 2011

Decoding Ipsilateral ECoG Signals, Liu, M.S. 2011