A HIERARCHICAL REPRESENTATION AND DECODING OF FMRI DATA BY PARTITIONING A BRAIN NETWORK

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HAZAL MOĞULTAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2017

Approval of the thesis:

A HIERARCHICAL REPRESENTATION AND DECODING OF FMRI DATA BY PARTITIONING A BRAIN NETWORK

submitted by **HAZAL MOĞULTAY** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Gülbin Dural Ünver _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering**

Prof. Dr. Fatoş T. Yarman Vural _____
Supervisor, **Computer Engineering Department, METU**

Examining Committee Members:

Prof. Dr. Göktürk Üçoluk
Computer Engineering Department, METU _____

Prof. Dr. Fatoş T. Yarman Vural
Computer Engineering Department, METU _____

Prof. Dr. Nafiz Arıca
Computer Engineering Department, Bahçeşehir University _____

Assoc. Prof. Dr. Ahmet Oğuz Akyüz
Computer Engineering Department, METU _____

Assist. Prof. Dr. Murat Perit Çakır
Department of Cognitive Science, METU _____

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: HAZAL MOĞULTAY

Signature :

ABSTRACT

A HIERARCHICAL REPRESENTATION AND DECODING OF FMRI DATA BY PARTITIONING A BRAIN NETWORK

Moğultay, Hazal
M.S., Department of Computer Engineering
Supervisor : Prof. Dr. Fatoş T. Yarman Vural

September 2017, 64 pages

In this study, we propose a hierarchical network representation of human brain extracted from fMRI data. This representation consists of two levels. In the first level, we form a network among the voxels, smallest building block of fMRI data. In the second level, we define a set of supervoxels by partitioning the first level network into a set of subgraphs, which are assumed to represent homogeneous brain regions with respect to a predefined criteria. For this purpose, we develop a novel brain parcellation algorithm, called BrainParcel. As current literature tends to represent human brain as a graph, BrainParcel adopts this approach. The suggested algorithm partitions a brain network, called mesh network using a graph partitioning method. The supervoxels obtained at the output of BrainParcel form partitions of brain as an alternative to anatomical regions (AAL). Compared to AAL, supervoxels gather the linearly dependent voxels. As the next step, we form a mesh network among the supervoxels. Therefore, we represent fMRI data by two networks of different granularity. The first network is at voxel level, whereas the second is at supervoxel level.

In order to test the representation power of this two level network, we suggest an ensemble learning architecture, called Cognitive Learner. The suggested ensemble learning method is used in brain decoding problem, where we classified the cognitive states. The results applied on an object recognition problem show that the suggested BrainParcel algorithm together with Cognitive Learner has a better representation power on brain decoding in terms of classification accuracy.

Keywords: fMRI, Brain Segmentation, Mesh Model

ÖZ

FMRG VERİLERİNİN BEYİN AĞLARI ÜZERİNE KURULU HİYERARŞİK TEMSİLİ VE ÇÖZÜMLENMESİ

Moğultay, Hazal
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi : Prof. Dr. Fatoş T. Yarman Vural

Eylül 2017, 64 sayfa

Bu çalışmada, insan beyninin modellenmesi için, fonksiyonel Manyetik Rezonans Görüntüleme (fMRG) verileri kullanılarak oluşturulan iki seviyeli hiyerarşik bir ağ modeli öneriyoruz. İlk seviyede, fMRG verilerinin yapıtaşı olan vokseller üzerinde bir ağ oluşturuyoruz. İkinci seviyede ise, ilk seviyedeki ağ yapısını alt ağlara parçalayarak, homojen beyin bölgeleri elde ediyoruz. Süper voksel adını verdiğimiz bu bölgeleri elde etmek için BrainParcel adını verdiğimiz bir beyin bölütleme algoritması geliştirdik. Son yıllarda insan beynini çizge kuramı kullanarak modellemek önem kazandığı için, BrainParcel de bir çizge parçalama yöntemidir. Bu algoritma, Örgü Ağları Modeli kullanılarak oluşturulan bir beyin ağını parçalayarak beynin alt ağlarını bulmaktadır. Çıktı olarak elde edilen süper vokseller, anatomik beyin bölgelerine bir alternatif oluşturmaktadır ve onlardan farklı olarak doğrusal bağımlı vokselleri bir araya getirmektedir. İkinci seviyede, süper vokseller arası bir örgü ağı kurmaktayız. Böylece, fMRG verilerini ilk seviyesini vokseller arası, ikinci seviyesini ise süper

vokseller arası bir çizgenin oluşturduğu iki seviyeli hiyerarşik bir yapı içerisinde göstermekteyiz.

Bu hiyerarşik yapının bilişsel süreçlerisunum gücünü ölçmek amacıyla Cognitive Learner adını verdiğimiz bir topluluk öğrenme mimarisi önermekteyiz. Bu mimari, insan beyni deşifresi üzerine çalışmakta olup, bilişsel süreçleri sınıflandırmaktadır. Cognitive Learner bir nesne tanıma problemi sırasında elde edilen fMRG verileri üzerinde denenmiştir. Sonuçlar göstermektedir ki, BrainParcel Cognitive Learner ile birlikte kullanıldığında beyin çözümlemesi için sınıflandırma performansını arttırmaktadır.

Anahtar Kelimeler: fMRG, Beyin Bölütleme, Örgü Ağları

To the memory of my dearest grandfather,

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor Prof. Dr. Fatoş Tünay Yarman Vural. Ever since the times I was an undergraduate student, she has been a role model for me. She not only thought me the methods and ethics of an academical study, but also she has enhanced my perspective on life. I found myself very privileged for being her student. She is the architect of this study, and I owe her a lot for her support and encouragement.

I owe special thanks to my colleagues Arman Afrasiyabi, Emre Aksan, Sarper Alkan, Barış Nasır, Güneş Sucu, Burak Velioğlu and Ozan Yıldız for their support and friendship. Most of all, I am grateful to Dr. Itır Onal. She has been very encouraging and she played a very important role about laying the foundations of this study. I know she will be an excellent professor soon, I am already jealous of her future students.

I wish to thank my flatmate Dilek Koptekin for her never-ending patience. She has been there in all the ups and downs that I have faced through this study. But mostly, I am thankful to her for accepting me as a sister. It meant a lot!

I also would like thank Berkay Gürgen for his unflagging support and complete trust. Without him I would not even apply for a masters degree.

I would like to extend my deepest gratitude to my family who never lost faith on me. I am thankful to my dad, Hakan, who has always supported my actions and decisions. Also, there are no words to describe my gratitude to my mom, Pınar. Without her, I would never believe in myself and never accomplished this far. I have always felt her endorsement. As we always say; in the end, we will always have each other!

I acknowledge that this study has been supported by TÜBITAK (The Scientific and Technological Research Council of Turkey) project 114E045.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xi

LIST OF TABLES xv

LIST OF FIGURES xvi

LIST OF ALGORITHMS xix

LIST OF ABBREVIATIONS xx

CHAPTERS

1 INTRODUCTION 1

1.1 Problem Definition 2

1.2 Proposed Method 3

1.3 Contributions 5

1.4 Thesis Outline 6

2 BACKGROUND ON FMRI DATA REPRESENTATION METHODS AND BRAIN DECODING 9

2.1 Functional Magnetic Resonance Imaging (fMRI) Data Characteristics 9

2.2 fMRI Data Representation and Brain Decoding 11

2.3 Brain Parcellation 12

2.3.1 An Empirical Brain Parcellation Method: Automated Anatomic Labeling (AAL) 12

2.3.2 Clustering Methods used for Brain Parcellation . . 13

2.4 fMRI Data Representation and Brain Decoding with Mesh Model 16

2.4.1 Mesh Model 17

2.5 Classification 20

2.5.1 Ensemble Learning for Brain Decoding 21

2.6 Chapter Conclusion 22

3 A NOVEL BRAIN DECODING SYSTEM 25

3.1 Concept of neighborhood 26

3.1.1 Definition of spatial neighborhood 28

3.1.2 Definition of functional neighborhood 28

3.2 Constructing of Mesh Network for N-cut Clustering 29

3.3 BrainParcel 33

3.4 Constructing Meshes for Voxels and Supervoxels - Estimation of MADs 35

3.5 Classification 38

3.5.1 Brain Decoding by Supervoxels 38

3.5.2 Cognitive Learner 38

3.5.2.1 Ensemble Learning: Fuzzy Stacked Generalization (FSG) 39

3.6 Chapter Conclusion 40

4 EXPERIMENTS ON BRAIN PARCELLATION, HIERARCHICAL NETWORK REPRESENTATION AND BRAIN DECODING 43

4.1 Dataset 43

4.2 Performances of Brain Decoding by Mesh Network, Estimated Among Anatomical Regions 44

4.3 Performances for Brain Decoding with Classical N-cut Segmentation 45

4.4 Performances of Brain Decoding with Spatially Constrained BrainParcel 47

4.5 Performances for Brain Decoding of Classical N-cut with Functional Constraints 49

4.6 Performances of Brain Decoding with Functional Constrained BrainParcel 51

4.7 Comparative Results of BrainParcel 51

4.8 Comparative results with Cognitive Learner 53

4.9 Chapter Conclusion 55

5 CONCLUSION AND FUTURE WORK 57

5.1 Discussion on BrainParcel and Possible Future Directions . . 57

5.2 Discussion on Cognitive Learner and Possible Future Directions 59

REFERENCES 61

LIST OF TABLES

TABLES

Table 4.1 Overall 2-class classification results acquired from the MADs of supervoxels. These results suggest that Spatial BrainParcel gives higher performances on average since it provides spatial continuity and functional homogeneity within brain parcels. These are the best results for all experiment setups. 53

Table 4.2 2-class classification accuracies obtained from the network over supervoxels with $l_c = 10$, $p_m = 10$ and $l_m = 10$. 54

Table 4.3 2-class classification accuracies obtained from the Cognitive Learner with $l_c = 10$, $p_m = 10$ and $l_m = 10$. 54

LIST OF FIGURES

FIGURES

Figure 2.1 Left most circle indicates the time where the stimulus is given to the subject. Then in accordance with the activation, oxygen level increases and as the right most circle shows oxygen level drops rapidly and converges to a stability level. Taken from Uludag et. al. [31] Note that horizontal axis shows time and the vertical axis shows the fMRI intensity. . 10

Figure 2.2 An example of a mesh model formed around voxel v_i. Figure shows the formation of a mesh around a seed voxel v_i. First mesh is formed for one cognitive stimulus k and these MADs are concatenated for all samples in the bottom part. Final matrix for one sample have the size pK with p neighbors and K cognitive stimuli in total. 18

Figure 2.3 Part 2 of a basic example of a mesh model. Figure shows the resulting matrix achieved by forming meshes around each voxel and each cognitive stimulus. Resulting matrix have the size $pN - by - K$ with p neighbors, N voxels and K cognitive stimuli in total. 19

Figure 2.4 Fuzzy Stacked Generalization architecture. Each base layer feature space is fed into a different classifier and resulting class conditional probabilities are concatenated to form a new feature space to be classified with a meta layer classifier. 23

Figure 3.1 Visual representation of case 1. IF $v_i \notin \eta_{v_j}^{p_c}$ AND $v_j \in \eta_{v_i}^{p_c}$ THEN $e_{i,j} = a_{i,j}$. 32

Figure 3.2 Visual representation of case 2. IF $v_i \in \eta_{v_j}^{p_c}$ AND $v_j \in \eta_{v_i}^{p_c}$ THEN we need to consider the correlation between two voxels. IF $CORR(v_i, v_j) \geq 0$ THEN $e_{i,j} = max(a_{i,j}, a_{j,i})$. IF $CORR(v_i, v_j) < 0$, THEN $e_{i,j} = min(a_{i,j}, a_{j,i})$. 33

Figure 3.3 Detailed architecture of Cognitive Learner. First layer consists of the voxel intensity values and second layer is the average brain segments. At each layer relationships among voxels/super-voxels is computed. FSG is used to fuse information from the two layers. 39

Figure 3.4 Detailed architecture of Fuzzy Stacked Generalization. Each base layer feature space is fed into a different classifier and resulting class conditional probabilities are used to form a new feature space to be classified with a meta layer classifier. 40

Figure 4.1 Accuracies obtained by the usage of anatomical regions. Leftmost bar shows the accuracy obtained by classifying the raw signals. The rest of the bars show the classification accuracies of mesh model for different neighborhood and regularization parameters. 45

Figure 4.2 Classification accuracies of classical n-cut, showing that mesh model with the regularization parameter, $l_m = 10$ gives highest accuracies regardless of the number of supervoxels and number of neighbors, p_m 46

Figure 4.3 Classification accuracies of spatial BrainParcel, showing that mesh model with $l_m = 10$ gives highest accuracies, regardless of the number of supervoxels and p_m. 48

Figure 4.4 Classification accuracies of classical n-cut with functional constrained neighborhood, showing that mesh model with $l_m = 10$ gives highest accuracies regardless of the number of brain parcels and p_m. . . . 50

Figure 4.5 Classification accuracies of Functional BrainParcel, showing that mesh model with $l_m = 10$ gives highest accuracies regardless of the number of supervoxels and p_m. 52

LIST OF ALGORITHMS

ALGORITHMS

Algorithm 3.1 Finding the Spatial Neighborhood 27

Algorithm 3.2 Finding the Functional Neighborhood 29

Algorithm 3.3 Classical Method for Brain Graph Construction 30

Algorithm 3.4 Undirected Graph Construction from Mesh Model 34

LIST OF ABBREVIATIONS

EEG	Electroencephalography
MEG	Magnetoencephalography
PET	Positron Emission Tomography
MRI	Magnetic Resonance Imaging
fMRI	functional Magnetic Resonance Imaging
BOLD	Blood Oxygen Level Dependent
N-cut	Normalized Cut
CL	Cognitive Learner
MVPA	Multi-voxel Pattern Analysis
HRF	Hemodynamic Response Function
PCA	Principal Component Analysis
AAL	Automated Anatomic Labeling
ROI	Region of Interest
MM	Mesh Model
SMM	Spatial Mesh Model
FMM	Functional Mesh Model
MADs	Mesh Arc Descriptors
FSG	Fuzzy Stacked Generalization
SVM	Support Vector Machine

CHAPTER 1

INTRODUCTION

"I don't know how to tell,
How to tell my sorrow!"
"Bilmem ki nasıl anlatsam;
Nasıl, nasıl size derdimi!"
-Orhan Veli Kanık

Humankind has shown a great interest to the nature from the beginning of his/her existence. Curiosity towards natural events drive them to observe and imitate these processes. As a consequence, successive developments took place and formed a basis for todays technology as we know it. Despite all realizations of nature through the years, there is still information regarding biological structures that remain as a mystery. Functional pathways in the human brain are one of these mysteries. Intelligence of humans stems from the complex structure of his/her brain, and yet we are still not capable of explaining the complete structure of our mind.

Early experiments of neuroscience studies were merely consisted of examining the physical form of the brain. Shape, size and fold structures were analyzed to understand various functions such as memory, vision and intelligence. Physical characteristics of the brain has been an interest for many researchers. For instance, Einsteins' brain has been removed from his body right after his death, without any notification to his family. Even though the action stands as an ethically controversial issue, it shows the interest of the people to the structure of a great mind. Even after many years, scientists were still examining his brain [7].

Despite all information hidden in the physical features of the brain, advances in the technology brought a new aspect to the topic. Today, we can observe the functional pathways of the brain on different scales. There are different brain imaging techniques to monitor the behavior of an operational brain. While techniques like EEG (Electroencephalography) and MEG (Magnetoencephalography) measures the electrical activity of the brain, PET (Positron Emission Tomography) gives information about the radioactive changes within the brain and, techniques such as MRI (Magnetic Resonance Imaging) and fMRI (functional Magnetic Resonance Imaging) can measure the changes in BOLD (Blood Oxygen Level Dependent) signals.

Decoding the brain activities during a cognitive task will result in many ground breaking results. When the actual mechanisms of the brain are analyzed, human computer interfaces can be built accordingly and the life of many will be a lot easier.

Although there are still unknown patterns related to the reactions in the human brain, many scientists are working on the issue and every new study gets us one step closer to mind reading, the remedy of one of the greatest mysteries in the nature.

1.1 Problem Definition

On the purpose of mind reading, brain decoding studies become prominent. Brain decoding involves composing a mathematical model on fMRI data which is basically a collection of signals recorded while the subject is performing a cognitive task for a limited time. Each task corresponds to one of multiple classes, and during these cognitive tasks, fMRI machine generates a brain volume of approximately 20.000 - 100.000 voxels at each time instant. Brain decoding aims to classify the underlying cognitive tasks from the fMRI recordings by a computational model. Concatenation of the signal intensities of voxels constitute the feature space for the classification.

These studies have a major drawback that they all suffer from the curse of dimensionality problem. Since the size of the feature space is too high, it is difficult to process the data and mathematically represent the cognitive tasks. Moreover, noisy

structure of data makes it even harder to capture the corresponding activity through a cognitive activity. Therefore, feature elimination, dimension reduction and region selection methods are widely used for fMRI datasets. Clustering the feature space into meaningful regions such that each region resembles a cognitive process, is one of the popular methods used in the literature for this purpose, which is also the field of occupation of this study.

1.2 Proposed Method

In this study, we attempt to generate a computational network model of human brain which represents the information embedded in fMRI data in a compact way, while avoiding the curse of dimensionality problem. We propose a hierarchical network representation of human brain for brain decoding called Cognitive Learner. In the first level, we form a network among voxels. In the second layer, we generate a brain parcellation and generate super-voxels with the novel algorithm that we propose, called BrainParcel. At both layers, we extract dependencies among voxels/supervoxels and use Cognitive Learner architecture for brain decoding to classify cognitive stimuli based on the obtained dependencies.

From the brain parcellation perspective, there exists a variety of methods [29, 14], among which we focus on the normalized cut (N-cut) algorithm since it is used to analyze the data in a graph theoretical approach. N-cut represents each voxel, which is the smallest volume fMRI machinery acquires information from and forms the feature space, as nodes of a brain graph rather than individual entities. Previous studies, calculate edge values among voxels as the correlation between the intensity values of two voxel nodes. In this study, we calculate edges from mesh model which has been used previously for classification of cognitive states.

Onal et al. showed that using the relations among voxels a set of voxels has more representational power than raw intensity values of voxels. They proposed mesh model which calculates the interdependencies of a voxel to its neighbors [19, 20, 21]. They select a voxel, called the seed voxel, and define its neighbors. For each cognitive

stimuli through the time course, the intensity of the seed voxel is represented as the weighted linear combination of the intensities of its neighbors. Therefore, they form a local network around the seed voxel, where the weights of the linear combination represents the arc weights of the network. These arc weights represent the dependencies between the seed voxel and its neighbors. Concatenated edge weights are used as features for classification. In BrainParcel, rather than using the correlation as edge weights between voxel pairs, we have used a set of mesh networks to create a brain graph. Unlike the original study, in which meshes are formed for each cognitive stimulus, we generated meshes around each voxel for the entire time course. Since the mesh around each seed voxel is calculated independently, acquired graph is directional, which means there are two directed edges between two voxels. In order to prevent directionality, we used a thresholding metric such that if the correlation between the two voxels is positive, we select the higher edge value, and we select the lower otherwise. Thus, we assign a single weight for each edge, and used n-cut to partition this brain graph.

Notice that, neighbors of a seed voxel can be selected spatially or functionally, meaning that physical proximity, or the correlational similarity can be favored during the selection of neighbors respectively. Selection of the neighborhood definition, causes a spatial or functional constraints on the system such that resulting brain parcels contain voxels that are spatially or functionally closer to each other. With the intent of comparison, we have carried both methodologies.

One critical problem of parcellation is to identify the number of brain regions a priori. Increasing the number of parcels results in smaller brain regions. In order to observe the effects of the region size, we obtain a different number of brain parcels. Each brain parcel is called a supervoxel. Later, we calculate the average time signal for each brain parcel and generate a single, representative intensity for regions.

At this point, we obtain two different representations of the human brain, one is the raw voxel intensity values, and the other is the average intensities of the supervoxels. We performed the original mesh model for classification purposes. We formed meshes around each voxel/supervoxel for each cognitive stimulus and formed two

networks of different granularities. These networks are assumed to represent different characteristics of the activations in the brain. Information obtained from these networks are fused together in an ensemble learning architecture which we all Cognitive Learner for brain decoding purposes.

To sum up, we propose a hierarchical machine learning architecture, called Cognitive Learner (CL), for decoding brain states using fMRI data. The architecture is composed of two layers. First level, called voxel level, represents a cognitive state by a high resolution mesh network, formed among the voxels of fMRI data. At the second level, cognitive states are represented by a coarse resolution mesh network. In order to compose the second level network, we use BrainParcel to partition the network obtained from the voxel level and generate supervoxels. Then, we form another mesh among supervoxels. The arc-weights of mesh networks at each level are estimated using mesh model, and an ensemble learning algorithm is used to fuse the information obtained from two levels.

1.3 Contributions

- In the literature of brain parcellation, n-cut algorithm is conducted on a graph where correlation value between node pairs are assigned as edge weights. In the proposed method, we compute edge values of a set of local meshes which is shown to be better in terms of representing the relationships among voxels. We argue that composing the brain graph with high representation power edge values yields better brain parcels in terms of distinguishing cognitive tasks.

- Partitioning the physical voxel space to obtain a brain parcellation may necessitate different constraints. Selecting the parcels among spatially or functionally closer voxels is a common procedure. In this study, we provide both functionally and spatially constrained options and compare them based on the classification accuracy.

- We provide an experimental comparison of both functionally and spatially constrained models against their corresponding counterparts from the literature.

- Various studies form brain networks among the predefined brain parcels, called Automated Anatomic Labeling (AAL) [30]. In this study, we argue the drawbacks of this method and compare it with the network over supervoxels generated at the output of BrainParcel.

- We propose a new two layered learning architecture that fuses information from network of voxels and the network of supervoxels and increases the representational power of individual networks.

Cognitive Learner part of this thesis has appeared on the following publication;

- H. Mogultay, and F. T. Y. Vural. "Cognitive learner: An ensemble learning architecture for cognitive state classification." In Signal Processing and Communications Applications Conference (SIU), 2017 25th. IEEE, 2017.

1.4 Thesis Outline

Chapter 2 provides a brief introduction to the available solutions and methods for brain parcellation and decoding in the literature. Starting with the characteristics of the fMRI imagery, this chapter covers representation of anatomical regions, popular brain parcellation methods of fMRI data, network representation and brain decoding techniques. Deep analysis on mesh model is given since we focus on it.

In Chapter 3, the suggested BrainParcel and Cognitive Learner methods are introduced. Both spatial and functional mesh model is expressed and the idea to integrate mesh model with brain parcellation is covered. Then, detailed architecture offered for Cognitive Learner and classification procedure for brain decoding is explained.

Chapter 4, provides experimental results applied on BrainParcel and Cognitive Learner. Brain parcels obtained for all the covered methods, and the BrainParcel are compared in terms of classification accuracy. All parameters are explicitly given and their affects on the performances are examined.

Chapter 5 summarizes the overall study and discusses the outcomes of the Cognitive Learner. Also, possible future directions are proposed, to extend Cognitive Learner for brain decoding studies.

CHAPTER 2

BACKGROUND ON FMRI DATA REPRESENTATION METHODS AND BRAIN DECODING

"Let's look at the sky!"
"Göğe bakalım!"
-Turgut Uyar

Aim of this chapter is to bring familiarity to functional magnetic resonance imaging (fMRI) studies. In this direction, first of all, important characteristics of fMRI imagery are briefly explained, concentrating on the problems encountered on machine learning aspects. Then, brain parcellation algorithms are taken into consideration and different methods are compared since they constitute the backbone of this study. Also, mesh model is covered as a network representation of fMRI data. Finally, classification methods, particularly the ensemble learning method, that is the subject of this study, is surveyed.

2.1 Functional Magnetic Resonance Imaging (fMRI) Data Characteristics

Advances in brain imaging techniques enabled us to measure the activation changes in the human brain. In terms of observation of cognitive dependencies, functional magnetic resonance imagery (fMRI) is one of the mostly used imaging techniques. As Pauling and Coryell explained, fMRI imagery uses magnetic resonance imaging (MRI) machines. In principle, oxygen dependencies of neuron groups fluctuate in accordance with the activation, and MRI machines can detect those changes through the

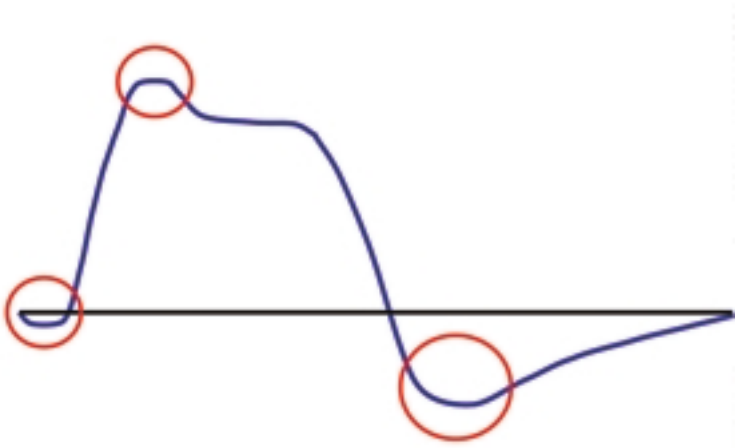


Figure 2.1: Left most circle indicates the time where the stimulus is given to the subject. Then in accordance with the activation, oxygen level increases and as the right most circle shows oxygen level drops rapidly and converges to a stability level. Taken from Uludag et. al. [31] Note that horizontal axis shows time and the vertical axis shows the fMRI intensity.

scan [24]. By a controlled group of stimulus, fMRI measures the cognitive activities of the subject during the scan. [24]. Uludag et. al. explained that while the subject stays in the fMRI machine, brain activity is manipulated by the given stimulus. Scans can be collected through the resting state, where subjects are asked to stay in a relaxed mental position, or during a predefined task. Those tasks can be selected from a variety of cognitive tests such as, listening a music or watching a video as well as solving a mathematical problem or matching images. Each task activates related parts of the brain. Responsible neuron groups activate with the given task and their oxygen consumption increases with the activation. Initially increasing oxygen level, falls through the consumption and converges to a stability level. fMRI machines measure this oxygen behaviors called Blood Oxygen Level Dependent (BOLD) signals per voxel. Figure 2.1 shows this behavior, namely the hemodynamic response function (HRF) [31] where horizontal axis is time and the vertical axis is the fMRI intensity.

Each voxel, which is the smallest block of fMRI imagery, is a volume around $1mm^3$ and contains thousands of neurons. Number of voxels in an fMRI dataset is very large and creates a huge feature space for brain decoding tasks. There are around 20.000

to 100.000 voxels depending on the characteristics of fMRI machine. Also, fMRI requires great amount of afford and time to collect samples, and subjects can be hold within the machine for a limited time. Therefore, number of cognitive stimuli is quite low when compared to the number of voxels which constitutes the feature space for brain decoding studies.

2.2 fMRI Data Representation and Brain Decoding

Multi Voxel Pattern Analysis (MVPA) studies has a leading role in the brain decoding literature. MVPA involves recognizing and analyzing the cognitive states represented by voxel intensity values of fMRI data, using machine learning techniques. A feature space is extracted from voxel time series using various methods, such as concatenating the maximum values of voxel intensity values recorded during each cognitive task. This approach creates a feature space of dimension D, where D is the total number of voxels in a brain volume. However, recall that fMRI data contains a large feature space with less number of samples to work with, scientists use different dimension reduction techniques for feature extraction on the data such as, principal component analysis (PCA) or voxel selection.

One of the most common methods is to reduce dimensionality with Principal Component Analysis (PCA). Objective of PCA is to reduce dimensions by mapping the samples to a new feature space while increasing the variances among features. For this purpose, a new space with less dimensions is created through the eigenvectors. Eigenvectors with higher eigenvalue has the most variance among them and they are more powerful than others in terms of data representation [8]. However, altering the feature space is sensitive to the noise. Important information through some features may be lost during this change. Therefore, PCA is not a reliable dimension reduction technique for fMRI studies.

The high dimension of the feature space not only brings the severe dimensionality curse problem, but, it also increases the computational complexity of the learning algorithms. In order to solve these problems, voxel selection methods are used to

eliminate the redundant information carried in the fMRI signal. Afrasiyabi et. al. showed that using a small subset of voxels is sufficient to distinguish different classes [1]. The main problem with voxel selection is to lose important information about the underlying cognitive process. In case of excessive selection, informative voxels might be eliminated.

2.3 Brain Parcellation

Necessity towards a compact representation, compelled scientists to split the available future space into meaningful regions and represent each region individually, as an alternative to voxel selection. In terms of brain parcellation, meaningful regions represent voxel groups such that within each group voxels react similarly to a cognitive stimulus. Observing the natural pathways within the brain, one can detect that neurons work in dependence with each other and in a harmony rather than standalone chemical activations. In parallel with this observation, we can conclude that voxels, which are basically just neuron groups, also work together through a cognitive task. Assuming those voxels give similar responses to a stimuli, grouping them together for a cognitive process is a rational strategy. Biologically defined anatomical atlases and well-defined clustering algorithms are widely used for this kind of grouping tasks.

2.3.1 An Empirical Brain Parcellation Method: Automated Anatomic Labeling (AAL)

Anatomical brain regions are defined by experimental neuroscience and they are called Automated Anatomic Labeling (AAL) regions. They have been used as a generic brain parcels. There are 116 predefined regions in the standard brain atlases and each region can be considered to contain dependent voxel groups that contribute similarly to a cognitive task [30]. Anatomical regions are defined in the well-accepted brain imagery preprocessing toolbox called SPM. Our datasets are also preprocessed in this toolbox. Penny et. al. explains related information about the underlying methodology in their book [25].

In order to reduce the dimensionality of the feature space, representative signals can be selected per each anatomical region. Average time series are used as a representative signal for anatomical regions. Average signal is treated as the super-voxel and implementations are carried for a smaller feature space that constitutes the super-voxels [2, 26]. Size of the feature space is equal to the number of brain regions. Other than that, standard brain atlases can be used for Region of Interest (ROI) selection mechanisms for a group of cognitive tasks. Since different cognitive tasks are associated with different anatomical regions, selecting the relatively active part of the brain is adequate for representing the task related signal structures [11, 20, 21].

Standard brain atlases are preferred, since they provide cross subject parcellation with neuroscientific dependencies. Unfortunately, anatomical regions are specified regardless of the cognitive task and this approach loses the subject-specific and task dependent information, and sometimes it may be misleading to focus on a specific brain region. Besides, sizes of the regions vary extremely across the subjects and distribution of voxels are not homogeneous within a region. On these grounds, forming regions with subject and task dependent borders is required.

2.3.2 Clustering Methods used for Brain Parcellation

In order to partition the voxels into a set of homogeneous regions, well defined clustering methods such as k-means [9, 10, 17], hierarchical clustering [2, 4], and spectral clustering [27, 32] are suggested in the literature. Main goal is to create brain regions consisting of voxels that give similar responses to a cognitive task. Different requirements can be established. Some of the current studies, bring spatially closer voxels together considering only the location information in analogy with the anatomical regions [9, 10]. Underlying assumption of this studies is that if a group of voxels activate similarly to a cognitive task, their chemical activation should trigger each other and they must stand physically closer. Although this method improves the strict norms of anatomical regions, it still does not consider the fact that, there might be physically distant, correlated voxels. On the other hand, with the extending research on the area, there is a tendency to group functionally closer voxels in a brain parcel.

Observing distant voxels may behave similarly to a cognitive task, considering them in a same region provides functionally homogeneous brain parcels. In line with these requirements, there is growing support for the claim that brain regions should be both functionally homogeneous and spatially contiguous. Flandin et. al. used k-means with 3D geodesic distances and stated that any functional restrictions can be forced with weighted geodesic maps [10]. Craddock et. al., on the other hand, used n-cut on a spatially constrained brain graph with functional edges [6]. Similarly Wang et. al. used n-cut and combined n-cut with simple linear iterative clustering (SLIC) for clustering [33]. Blumensath et. al. used region growing for brain segmentation with functional metrics and spatial constraints between samples [3]. Bellec et. al. also used region growing with functional metrics within the 26 spatial neighborhood in 3-Dimensional space. Background on neuronal activity also supports this idea such that physically closer neurons are in chemical interaction with each other and this interaction can be interpreted as the functional similarity.

With these objectives in mind, many different clustering algorithms are applied to create data dependent homogeneous brain parcels. Well accepted clustering algorithms such as k-means, hierarchical clustering and spectral clustering are practiced and their advantages and disadvantages are revealed on variety of datasets [29, 14]. While parcellating the voxels according to a predefined distance measure, these algorithms can all achieve spatial or functional similarity. Commonly, voxels are partitioned according to Euclidean or Manhattan distance when the locality is concerned. On the other hand, functional similarity can be achieved by using the Pearson correlation function given in the following equation;

$$corr(X, Y) = \frac{cov(X, Y)}{\sigma(X)\sigma(Y)}, \tag{2.1}$$

where $cov(X, Y)$ indicates the covariance between X and Y, and $\sigma(X)$ and $\sigma(Y)$ indicates the standard deviations of X and Y respectively. While all these algorithms are widely accepted, they still cannot serve the purpose of achieving both spatial and functional restrictions on their own. In order to solve this problem, scientists combine different methodologies such that one part fulfills spatial dependencies, the

other fulfills functional [6, 33]. Algorithms that represent the set as a graph such as normalized cut (N-cut) can meets this expectation [28]. From brain parcellation perspective, n-cut is a well-studied method [32]. Also, among other graph theoretic methods, n-cut is shown to be better in terms of brain region qualifications[27].

Normalized Cut (N-cut) is a graph theoretic segmentation algorithm first proposed for image segmentation problems. Algorithm represents each sample as a node of a graph and splits the graph into predefined number of subgraphs in an iterative manner. Resulting subgraphs constitute the intended segments.

N-cut algorithm has an approach with two steps. In the first step, all the samples, voxels in the brain parcellation case, are represented as a graph $G = (V, E)$, where the set of samples $V = [v_1, v_2 \dots v_N]$ are the nodes of the graph, and $E = [e_{1,1}, e_{1,2}, \dots e_{N,N}]$ are the edges. Edge values represent the similarity between two nodes. The literature shows a consensus on using the correlation as the edge value as follows:

$$e_{i,j} = corr(v_i, v_j), \tag{2.2}$$

where $corr(v_i, v_j)$ calculates the correlation between two voxels according to their covariance and standard deviation by equation 2.1.

As the second step of N-cut algorithm, edges of the graph G is eliminated until the number of the connected components of the reduced graph met the number of clusters. Elimination is executed in an iterative manner. At each iteration, an edge which minimizes Equation 2.3 is eliminated. Here V represents the set of all nodes and A and B are the two remaining connected components obtained by removal of an edge.

$$N_cut = \frac{\sum_{v_i \in A, v_j \in B} e_{i,j}}{\sum_{v_i \in A, v_n \in V} e_{i,n}} + \frac{\sum_{v_i \in A, v_j \in B} e_{i,j}}{\sum_{v_j \in B, v_n \in V} e_{j,n}}. \tag{2.3}$$

Algorithm halts, when the number of connected components in the graph G converges to a predefined number of clusters. Nodes of each connected component is returned as members of a cluster.

Craddock et. al. revised this algorithm to restrict the resultant brain parcels to be spatially contiguous and functionally homogeneous [6]. Original n-cut defines an edge between each pair of voxels. Rather than that, they allow edges to exist between physically neighboring voxels only. In other words, they modified the edge definition as follows;

$$e_{i,j} = \begin{cases} corr(v_i, v_j) & , dist(v_i, v_j) \leq d_t \\ 0 & , otherwise, \end{cases} \tag{2.4}$$

where $dist(v_i, v_j)$ in Equation 2.4 is Euclidean distance between samples v_i and v_j. d_t is selected to be $\sqrt{3}$, such that only the edges among 26 physical neighbors are allowed, which forces n-cut to create spatially continuous clusters. Since the edge values are correlation values, functional homogeneity is achieved within clusters. Moreover they removed the edges with correlation values less than 0.5 in order to reduce the weak connections.

2.4 fMRI Data Representation and Brain Decoding with Mesh Model

Recall that, MVPA studies decodes the cognitive states through raw fMRI signals. Voxel intensity values acquired from fMRI imagery composes a large feature space. With MVPA, classification is carried on this space without altering it. These studies played a lead role in the fMRI literature [5, 16, 18]. MVPA is a supervised method where a set of samples are separated for training and the rest is kept for testing. A classifier model is built on top of the training set and the related parameters are learned. MVPA models are used for diagnosis detections studies as well as cognitive state classification studies and it has shown that it is a powerful approach in terms of representation of brain states.

However, with the increasing number of available datasets, conventional MVPA model became insufficient for modeling cognitive states of the brain. Knowing the biological structures of human brain, that neurons work in an interactive manner, recent studies focused on the relations among voxels rather than individual signals. Ozay et. al, demonstrated this idea suggesting mesh model which is a graph structure that

identifies the links among voxels [22].

2.4.1 Mesh Model

Mesh Model (MM) represents intensity values of voxels as a weighted linear combination of its neighboring voxels, defined on a neighborhood system. Each voxel is handled individually and represents a node in the overall model. A star mesh is formed around each voxel and its p neighbors. In each mesh, voxel at the center is called seed-voxel and the rest of them are called neighbors. p nearest neighbors of voxel v_i for sample k are shown as $\eta^p_{v_i(k)}$ and they can be selected spatially or functionally such that, spatial neighbors has a smallest Euclidean distance with the seed-voxel whereas functional neighbors has maximum functional similarity. Meshes are formed using the full length time series for voxels. Assuming s measurements are taken per each cognitive stimulus, time series of a voxel v_i for stimuli k is a s dimensional vector shown as $v_i(k) = [v_i(k)^1, v_i(k)^2, \ldots v_i(k)^s]$.

Spatial Mesh Model (SMM) selects the neighbors according to the physical distances among voxels in 3-Dimensional space by Euclidean distance [19, 21]. On the other hand, Functional Mesh Model (FMM), proposed by Onal et. al., selects functional neighbors with the highest p-correlation values obtained by Equation 2.1 [19, 20]. Afterwards, time series of the seed voxel is represented as a weighted combination of its neighbors by the following equation for each cognitive stimuli:

$$v_i(k) = \sum_{v_j(k) \in \eta^p_{v_i(k)}} a_{i,j,k} v_j(k) + \epsilon_{i,j}, \tag{2.5}$$

where $\eta^p_{v_i(k)}$ is the p nearest neighbors of voxel v_i for sample k and $a_{i,j,k}$ are the arc weights of the mesh network between the voxels and they are called Mesh Arc Descriptors (MADs) in classification. MADs are estimated using regularized Ridge regression method by the minimization of error term $\epsilon_{i,j}$. Ridge regression solves the

following equation in order to find arc weights of Equation 2.5;

$$a_{i,j,k} = (X'X + \lambda I)^{-1} X' v_i(k), \tag{2.6}$$

where X is a matrix formed by the concatenation of all $v_j(k)$ in the neighborhood of $\eta^p_{v_i(k)}$. λ is a constant value added to all diagonal entries of $X'X$. Entries $a_{i,j,k}$ are found by the optimization of the square of the error term $\epsilon_{i,j}$ by the following optimization function where outer summation repeats for each element of the vector $v_i(k)$;

$$min\left(\sum \left(v_i(k) - \sum_{v_j(k) \in \eta^p_{v_i(k)}} a_{i,j,k} v_j(k) \right)^2 + \lambda \sum_{v_j(k) \in \eta^p_{v_i(k)}} \left(a_{i,j,k} \right)^2 \right) \tag{2.7}$$

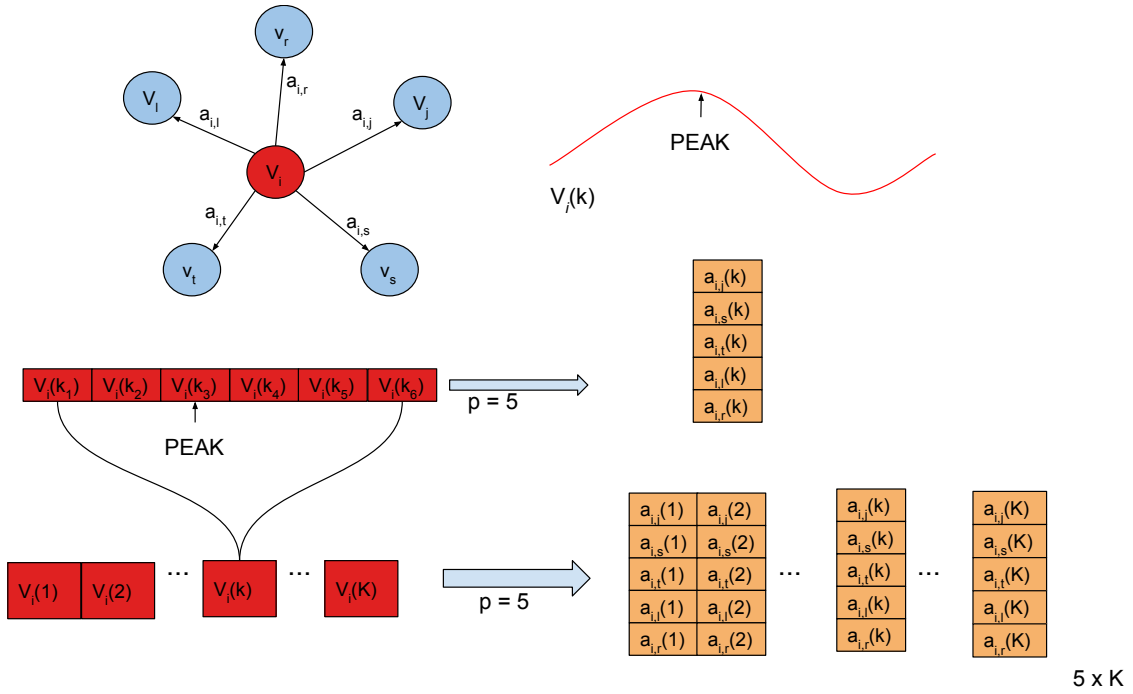


Figure 2.2: An example of a mesh model formed around voxel v_i. Figure shows the formation of a mesh around a seed voxel v_i. First mesh is formed for one cognitive stimulus k and these MADs are concatenated for all samples in the bottom part. Final matrix for one sample have the size pK with p neighbors and K cognitive stimuli in total.

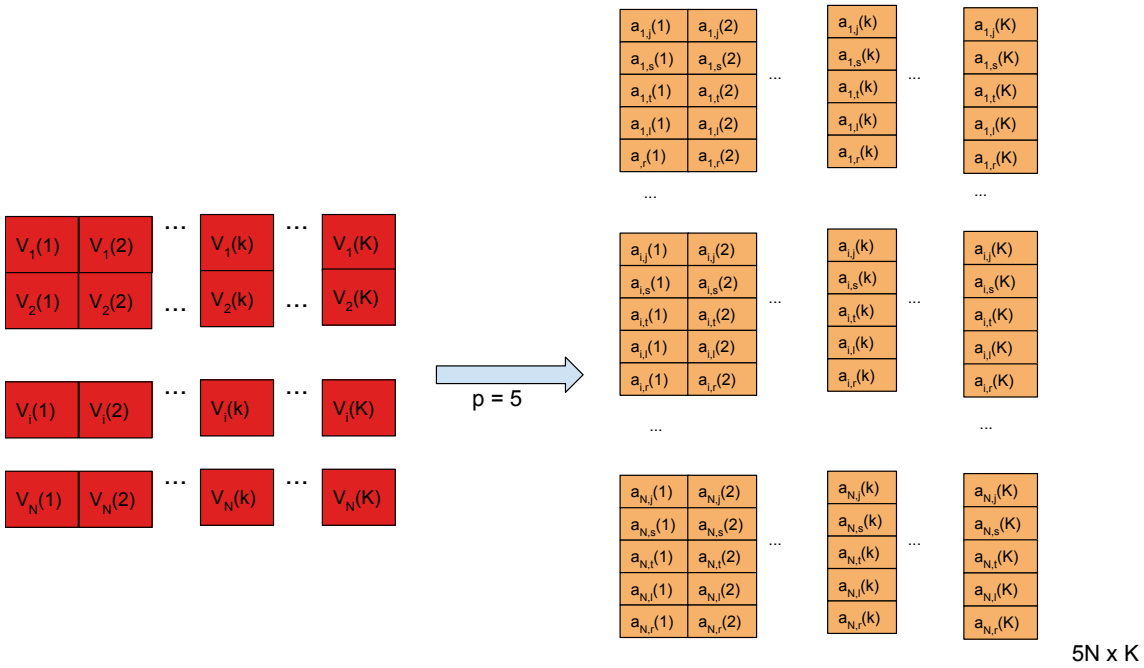


Figure 2.3: Part 2 of a basic example of a mesh model. Figure shows the resulting matrix achieved by forming meshes around each voxel and each cognitive stimulus. Resulting matrix have the size $pN - by - K$ with p neighbors, N voxels and K cognitive stimuli in total.

Assuming p neighbors, response of the seed voxels for a cognitive task is represented by the concatenation of mesh ark weights as a p dimensional vector. Proceeding with the idea, voxel respond forms a $1 - by - pK$ vector with K cognitive stimuli in total. Feature space is constructed by concatenating all the arc weights of meshes formed around voxels. Classification model is built on top of this feature space.

Consider the simple example shown on Figures 2.2 and 2.3. Figure 2.2 represents the formation of a mesh around a seed voxel v_i, red circle, for k^{th} sample with number of neighbors $p = 5$. Notice that, if we were to use raw signal intensities for classification, we should have used peak value. For one cognitive stimulus, mesh model forms a column matrix of length p. Considering all of the stimuli for one voxel, these column vectors are concatenated such that number of columns match the number of stimuli in the experiment, therefore the resulting matrix would have the size of $p - by - K$

assuming K is the number of stimuli. Figure 2.3 represents the final MADs matrix formed by mesh model. Final matrix is composed of the arc weights for all samples and all voxels and have the size of $pN - by - K$ assuming N is the number of voxels.

Forming meshes around each voxel results in a Mesh Network over the brain. Assuming $M = (V, A)$ is the mesh network, V is the nodes of the network formed by the voxels, $V = [v_1, v_2, \ldots v_N]$ and, A is the adjacency matrix defining the edges of the network and formed by arc weights $a_{i,j}$ of the single meshes in its entries.

2.5 Classification

A wide range of machine learning methodology can be used for the purpose of cognitive state classification. This section introduces the idea behind using logistic regression classifier, since the orientation of this study concentrates on logistic regression.

Logistic Regression is a density estimation method to learn the parameters of a hypothesis function on labeled training samples, and tries to predict the class probabilities for unknown samples. Letting X be the training set with m samples, hypothesis function defined on the unknown parameter set θ is given as follows;

$$h_\theta(X) = \frac{1}{1 + e^{-\theta^T X}}, \tag{2.8}$$

where $h_\theta(X)$ is the sigmoid function; therefore, the output of logistic regression classifier for any sample yield the class conditional probabilities for that sample. Unknown parameter set θ is estimated such that selected parameters minimize the cost function given in Equation 2.9 where y is the class labels in a binary class classification problem which is also the case in the current study;

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)}). \tag{2.9}$$

Even if the calculation of $J(\theta)$ is similar to the linear regression problem, defini-

tion of $Cost(h_\theta(x), y)$ makes logistic regression non-convex unlike linear problems. $Cost(h_\theta(x), y)$ is calculated as follows;

$$Cost(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)), & \text{if y = 1} \\ -log(1 - h_\theta(x)), & \text{if y = 0} \end{cases} \tag{2.10}$$

Learned parameter set θ is used for labeling the unknown samples. Labeling operation is carried by thresholding on class conditional probabilities.

2.5.1 Ensemble Learning for Brain Decoding

In recent years, ensemble learning takes an important place in the fMRI classification literature [12]. Ensemble learning is a method that combines the information obtained from multiple different classifiers in order to achieve higher learning performance. Basically, different classifiers are trained on the dataset, and their learned hyperspace is used to train a meta layer classifier. In classical approaches, all voxels are assumed to operate in conjunction. However, cognitive tasks are completed in hierarchical manner in the human brain. To our knowledge, neurons perform tasks in small groups; then, the resultant signals coalesce to generate a resultant reaction. In order to mimic the biological processes of the human brain, ensemble learning methods are used for brain decoding.

Kuncheva et. al. have used ensemble learning for their study is [12, 13]. They compared different approaches of ensemble learning with a single-layer classification approach. Their results support the idea that Support Vector Machine (SVM) can perform better than the chance level. On the other hand, ensemble learning methods can produce higher classification rates than single-layer classification when SVM is used as a base layer learner. Besides these facts, they stated that the complexity of the ensemble learning for fMRI data is significantly higher than the single-layer approach. Martinez et. al. have also studied on ensemble learning, in terms of complexity [15]. They used ADABOOST, which is used with other classifiers to increase their performances by weighting the classification results at meta layer, instead of random forest.

ADABOOST is used to select relevant base layer segments. Since the selection reduces the number of base layer classifiers, complexity is tremendously reduced when compared to other ensembles.

Ozay et. al. proposed an ensemble learning algorithm, called Fuzzy Stacked Generalization (FSG) [23]. FSG, first, gathers different representations of the dataset such that each representation contains a different information about the samples. Partitioning the available feature space into smaller chunks can be considered for this purpose assuming that each chunk represents a different characteristic of the samples to be classified. Then, an independent classifier is trained within each chunk and class conditional densities are obtained at the output of each classifier. Assuming L many base layer chunks, and we train a classifier C_l within each chunk, we obtain class conditional densities $P_l(w_j|x_i)$ for each sample x_i and each class w_j. Afterwards, these densities are concatenated so that they form a new meta layer feature space, and a meta layer classifier c_{meta} is trained and fuses information obtained from base layer classifiers. Figure 2.4 represents the work-flow of the FSG algorithm. Each blue box labeled with $Base_l$ is a base layer representation, within each different classifier c_l is trained and class conditional probabilities $M_l = P_l(w_j, x_i)$ is obtained, and concatenated to form a meta layer feature space. FSG algorithm is proven to work on multi class image dataset [23]. Moreover Alkan et. al. has experimented FSG on an fMRI dataset. They have used anatomical regions to define base layer chunks.

2.6 Chapter Conclusion

This chapter focuses on familiarizing the reader to the important topics related to fMRI data and its use for brain decoding. Background information about the methods and concepts that serves as the baseline of this study is covered. First of all, structure of the fMRI data is given since the models should reflect the characteristics of the underlying data set. After that, brain parcellation is covered. As anatomical regions give the basic brain parcels, clustering algorithms achieve more robust parcels with subject and task dependencies. Regarding brain decoding, this chapter introduces

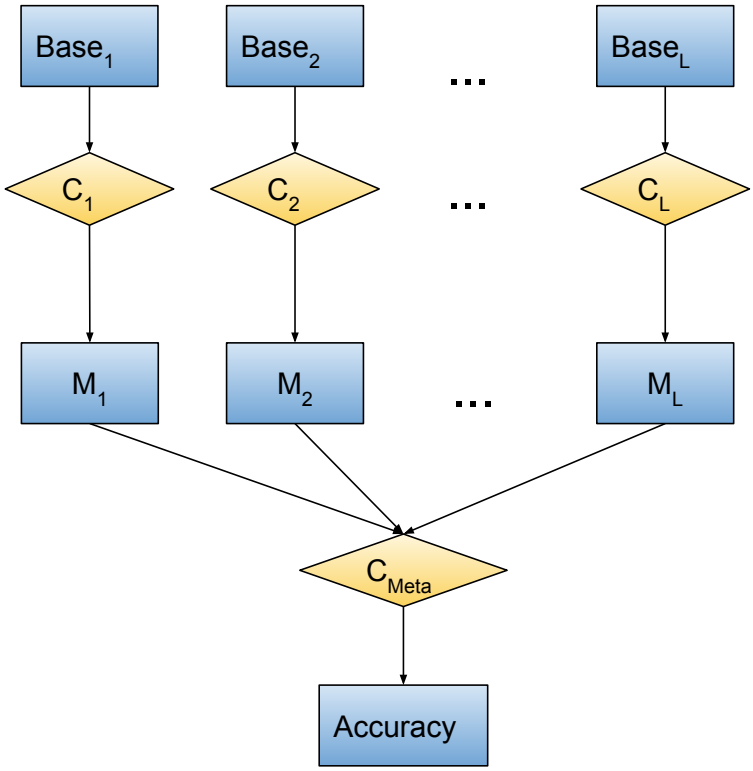


Figure 2.4: Fuzzy Stacked Generalization architecture. Each base layer feature space is fed into a different classifier and resulting class conditional probabilities are concatenated to form a new feature space to be classified with a meta layer classifier.

the concept of mesh model, which models the relationships among voxels and it has been sown to outperform other similarity measurements in terms of classification accuracy. In terms of classification, main classifier used in this study is explained and the ensemble learning architecture is covered.

CHAPTER 3

A NOVEL BRAIN DECODING SYSTEM

"Keep the flight in mind,
The bird may die"
"Kuş ölür, sen uçuşu hatırla"
-Füruğ Ferruhzad

Having the background information presented in the previous chapter, this chapter introduces the detailed information about the current study. Gathering the concepts of brain parcellation, and mesh model under the same roof, a compact brain parcellation method called BrainParcel is offered. BrainParcel is a revised n-cut clustering method which partitions a new graph, called Mesh Network. The network is formed by ensembling a set of local meshes, estimated from the fMRI data, independently. Originally, mesh model forms a local mesh around each voxel in a star topology. We offer a framework for combining these independent meshes to form a brain graph to represent a cognitive state. Then, with the use of n-cut, we partitions this graph into predefined number of subgraphs.

Our motivation is inspired from the biological structure of the human brain. As neuron groups works in a harmony, fMRI data should reflect this characteristic while representing the cognitive states of the brain. Onal et. al., adopted this idea and carried classification with mesh model [19, 20, 21]. In this study, we employ the mesh network for brain parcellation.

Moreover, we propose a two level hierarchical ensemble learning schema called Cog-

nitive Learner for cognitive state classification. First level consists of a fine network formed among the voxels, whereas the second level is represented by a coarse network formed among the supervoxels obtained at the output of BrainParcel. At each level, mesh model is used to extract the relationships among voxels and supervoxels, and an independent classifier is trained for each level. Outputs of each classifier is brought together in an ensemble learning architecture.

This chapter is organized as follows; first, graph structure for BrainParcel is explained. Both spatial and functional versions of the graphs are introduced. Later, necessary post-processing applied on these graphs are given. The purpose of the post-processing is to convert the acquired graph into n-cut requirements. Later, how to represent supervoxels, and how to create mesh architecture on top of the super voxels is explained. Through the chapter, while BrainParcel being explained, corresponding counterparts from the literature are also explained for convenience. Finally Cognitive Learner (CL) is introduced, and the suggested ensemble learning schema is analyzed in detail.

3.1 Concept of neighborhood

The concept of neighborhood takes an important place in this study. As illustrated in Chapter 2, several methods suggested in the literature define an edge between each pair of voxels in order to create a brain graph. In our study, we inspire from the biological structures of the human brain by means of chemical interactions, such that not all neurons activate each other. We try to transfer this knowledge to our brain parcellation model called BrainParcel. Background information on the topic leads us to define a neighborhood concept to model multiple connection of voxels.

Neighborhood of the i^{th} voxel v_i, is defined as the set of voxels that has a physical connection with v_i in the resulting brain graph. Assuming we define p_c many neighbors, neighborhood of v_i is represented by $\eta_{v_i}^{p_c}$. In other words, in the resulting graph that we feed to n-cut, there is an edge between v_i and the voxels within the set $\eta_{v_i}^{p_c}$. Note that number of neighbors p_c is a user defined parameter that can highly effect

the results.

Letting N be the number of voxels, we define an $N - by - N$ adjacency matrix ND to represent the neighborhood of voxels. Each entry of ND is calculated as follows;

$$ND(i,j) = \begin{cases} 1 & , v_j \in \eta_{v_i}^{p_c} \\ 0 & , otherwise. \end{cases} \tag{3.1}$$

i^{th} row of ND is shown by $ND(i,:)$, and represent the neighborhood of voxel v_i. Values of $ND(i,:)$ sums up to p_c since we define p_c number of neighbors.

$$\sum_{j=1}^{N} ND(i,j) = p_c \tag{3.2}$$

We define two different versions of the neighborhood. These two versions, namely spatial and functional neighborhood brings a spatial and functional constraint to the overall system as explained in the following subsections.

Algorithm 3.1 Finding the Spatial Neighborhood

```
1:  procedure SPATIAL_ND
2:      N ← number of voxels
3:      vXYZ ← coordinates of voxels
4:      p_c ← number of neighbors (user defined)
5:      distances ← ZEROS(N, N)
6:      ND_local ← ZEROS(N, N)
7:      for i := 1 → N do
8:          for j := 1 → N do
9:              distances(i,j) ← EUCLIDEAN DISTANCE(vXYZ(i), vXYZ(j))
10:     for i := 1 → N do
11:         indexes ← FIND p_c MINIMUM ENTRIES(distances(i,:))
12:         ND_spatial(i, indexes) ← 1
        return ND_spatial
```

3.1.1 Definition of spatial neighborhood

In the spatially constrained system, each voxel is connected to its physically closest p_c neighbors as Craddock et. al. proposed [6]. Therefore, $\eta_{v_i}^{p_c}$ is defined as the set of voxels, which has the p_c smallest Euclidean distance in 3-Dimensional space to voxel v_i. This approach ensures resulting clusters to be spatially contiguous.

Pseudo code of finding the spatial neighborhood, *Spatial_ND* is given at Algorithm 3.1. At line 9, Euclidean distances between voxels v_i and v_j is calculated according to their x, y and z coordinates in 3-Dimensional brain volume. Lines 11, 12 finds indices of the closest voxels to v_i, and updates the final neighborhood adjacency matrix (ND) accordingly. Encapsulating loop ensures ND to contain neighborhood information for all voxels. Notice that the resulting adjacency matrix will be symmetrical, since the spatial proximities cannot change among voxels.

3.1.2 Definition of functional neighborhood

In the functionally constrained system, voxels are allowed to make connection to others with similar time courses. Here, similarity is measured by Pearson Correlation given in equation 3.3. Two voxels are assumed neighbors only if their activation represented by time series is highly correlated. Each voxel is connected to its maximally correlated p_c neighbors. Therefore, $\eta_{v_i}^{p_c}$ is defined as the set of voxels, which has the maximum Pearson Correlation to voxel v_i. Notice that, Pearson correlation assigns a correlation value between -1 and 1, where high positive value means strong positive correlation.

$$corr(X, Y) = \frac{cov(X, Y)}{\sigma(X)\sigma(Y)}. \tag{3.3}$$

Pseudo code of functional neighborhood, *Functional_ND* is given at Algorithm 3.2. This algorithm is very similar to Algorithm 3.1 in terms of the information flow. The only difference is that in line 8, Pearson correlation is calculated among voxels, and at line 10 highly correlated neighbors are found.

Algorithm 3.2 Finding the Functional Neighborhood

procedure FUNCTIONAL_ND
    N $\leftarrow$ number of voxels
    $p_c \leftarrow$ number of neighbors (user defined)
    correlations $\leftarrow$ ZEROS(N, N)
    $ND_{functional} \leftarrow$ ZEROS(N, N)
    for $i := 1 \rightarrow N$ **do**
        for $j := 1 \rightarrow N$ **do**
            correlations(i,j) $\leftarrow$ PEARSON CORRELATION(v_i, v_j)
    for $i := 1 \rightarrow N$ **do**
        indexes $\leftarrow$ FIND p_c MAXIMUM ENTRIES$(correlations(i, :))$
        $ND_{functional}(i, indexes) \leftarrow 1$
    return $ND_{functional}$

It is very important to note that, the acquired graphs $ND_{spatial}$ and $ND_{functional}$ has a critical difference. While $ND_{spatial}$ is symmetrical, the other is not. In the case of spatial neighborhood, if voxel v_i is in the spatial neighborhood of v_j, $v_i \in \eta_{v_j}^{p_c}$; it is certain that v_j is in the neighborhood of v_i ;$v_j \in \eta_{v_i}^{p_c}$ since the Euclidean distance is used. On the other hand, it is not the case for functional neighborhood system. In other words, voxel v_i being in functional neighborhood of v_j, may not imply the spatial closeness.

3.2 Constructing of Mesh Network for N-cut Clustering

After identification of the neighborhood, a graph representation of the voxels is formed. N-cut performs on a graph $G = (V, E)$, where nodes are the voxels, $V = [v_1, v_2, \ldots v_N]$ with weighted undirected edges among them, $E = [e_{1,1}, \ldots e_{i,j}, \ldots e_{N,N}]$, such that $e_{i,j}$ represents the edge weight between voxels v_i and v_j . As explained in Chapter 2, current studies compute edge values by Pearson correlation and it ensures that the edge weights are indirected. We have employed this approach in order to compare the results to our method.

Algorithm 3.3 gives the pseudocode of the classical graph construction method based on Pearson correlation [6]. It requires the voxel intensity values and the neighborhood matrix that is explained in the previous sections. Edge weights of the graph G is calculated according to the correlation values between voxels according to the Equation 3.3. Unlike Craddock et. al. [6], we did not restricted correlation values to be higher than a threshold value, arguing that even the weak correlations may carry significant information related to the task. Lines 8 and 9 ensures the undirected structure of the graph is preserved regardless of the neighborhood system.

Algorithm 3.3 Classical Method for Brain Graph Construction

```
1: procedure ORIGINAL GRAPH CONSTRUCTION
2:     N ← number of voxels
3:     G ← ZEROS(N, N)
4:     for i := 1 → N do
5:         indexes ← FIND NONZERO ENTRIES(ND(i, :))
6:         v_indexes ← v_j where j ∈ indexes
7:         for j := v_indexes do
8:             G(i,t) ← CORR(v_i, v_t)
9:             G(t,i) ← CORR(v_i, v_t)
       return G
```

As opposed to the current studies, we form a brain graph with the help of mesh model which has been proposed by Onal et. al. [19, 20, 21]. The idea behind mesh model is to represent response of a voxel as the linear combination of its neighbors. Chapter 2 gives detailed information on the topic, in which task based meshes are formed around each voxel. On the other hand, we form meshes based on the complete time series recorded at each voxel.

fMRI imagery collects a time series for each voxel, when the subject is exposed to a cognitive stimulus. In the case of a block experiment design, which we have used, subjects are exposed to a stimuli for a specific time interval and the voxel responses over the entire brain volume are collected. Then, after a rest period, another stimulus is given to the subject. For each cognitive stimulus, 6 measurements are collected. Therefore, time series of a voxels response is a vector of 6 times the number of stimuli.

Response of the i^{th} voxel is shown by v_i. Based on the idea of mesh model, we represent each v_i as weighted average of other voxels in the neighborhood of v_i; $\eta_{v_i}^{p_c}$. Notice that Mesh Arc Descriptors (MADs) for classification are calculated per cognitive stimulus as explained in Chapter 2. However, we compute mesh arc weights from all of the voxels response, and call them arc weights rather than MADs. This representation is carried with a linear equation by the following formula;

$$v_i = \sum_{v_j \in \eta_{v_i}^{p}} a_{i,j} v_j + \epsilon_{i,j}. \tag{3.4}$$

As shown in Equation 3.4, weights of this representation are shown as $a_{i,j}$ and they are called the mesh arc weights estimated by Regularized Ridge Regression method by the minimization of error term $\epsilon_{i,j}$. Ridge regression solves an optimization function explained in Chapter 2, where the parameter λ rests as a regularization parameter.

Mesh arc weights forms the arc weights of resulting graph. While computing the meshes around voxels, we fill the edge values of the graph $G = (V, E)$. With each mesh, new set of edges are estimated, and G is updated accordingly. Recall that n-cut operates on undirected graphs, in which if there is an edge between any two nodes, there must be a scalar edge weight in between. However, mesh arc weights are directed. The reason is that, arc weights are calculated for each voxel independently. The most basic instance of this case is the non symmetrical structure of the functional neighborhood system. In order to solve this problem, a set of rules are adopted. These easy to understand rule set aims to relax the directed structure of the graph into undirected one. Simple thresholding method is practiced which is explained in the following part by an example.

Consider the case that the mesh is formed for the voxel v_i and v_j is in the neighborhood of v_i with mesh arc-weight $a_{i,j}$. Edge value $e_{i,j}$ is filled, based on the following rules.

- **Case 1 :** $v_i \notin \eta_{v_j}^{p_c}$ AND $v_j \in \eta_{v_i}^{p_c}$

  In this case, v_i is not in the neighborhood of v_j; therefore, the current value of

$e_{i,j}$ is set to zero. $e_{i,j}$ is simply updated to be $a_{i,j}$. Visual representation of this case is given at Figure 3.1.

- **Case 2 :** $v_i \in \eta_{v_j}^{p_c}$ AND $v_j \in \eta_{v_i}^{p_c}$

  In this case, v_i is in the neighborhood of v_j and the value $e_{i,j}$ has been set to $a_{j,i}$ beforehand. Then, the relationship between v_i and v_j is considered. Assuming highly correlated voxels should have a stronger edge between them, we employ the following thresholding method for this case;

  - $CORR(v_i, v_j) \geq 0$

    If v_i and v_j are positively correlated than the maximum arc weight is assigned as the edge value; $e_{i,j} = max(a_{i,j}, a_{j,i})$.

  - $CORR(v_i, v_j) < 0$

    If v_i and v_j are negatively correlated than the minimum arc weight is assigned as the edge value; $e_{i,j} = min(a_{i,j}, a_{j,i})$, as opposed to the previous case.

  Visual representation of this case is given at Figure 3.2.

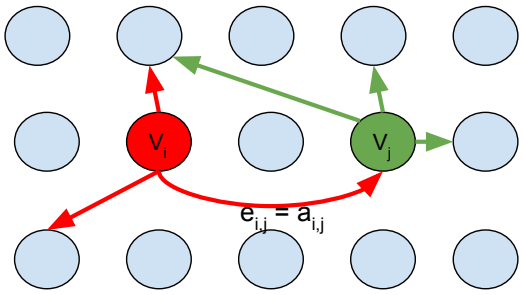


Figure 3.1: Visual representation of case 1. IF $v_i \notin \eta_{v_j}^{p_c}$ AND $v_j \in \eta_{v_i}^{p_c}$ THEN $e_{i,j} = a_{i,j}$.

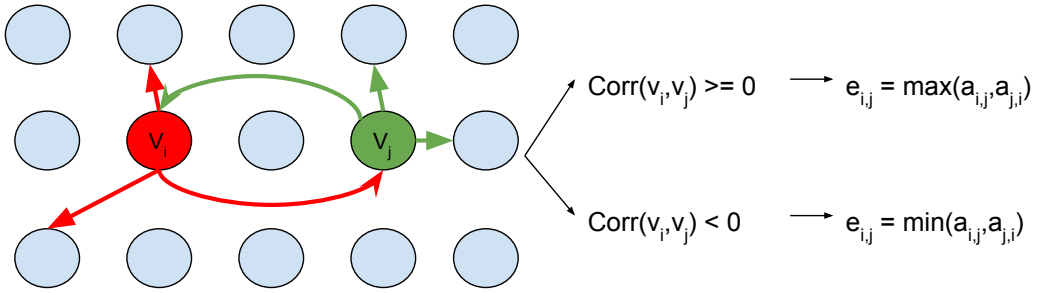


Figure 3.2: Visual representation of case 2. IF $v_i \in \eta_{v_j}^{p_c}$ AND $v_j \in \eta_{v_i}^{p_c}$ THEN we need to consider the correlation between two voxels. IF $CORR(v_i, v_j) \geq 0$ THEN $e_{i,j} = max(a_{i,j}, a_{j,i})$. IF $CORR(v_i, v_j) < 0$, THEN $e_{i,j} = min(a_{i,j}, a_{j,i})$.

Pseudo code of graph construction is given in Algorithm 3.4. Mesh construction and the graph filling is employed in an iterative manner. For each voxel, a mesh is formed around it, and the corresponding edges are filled accordingly. Algorithm requires the voxel intensity values and the neighborhood matrix extracted from Algorithm 3.1 or 3.2. For each voxel, its neighbors are found from the neighborhood matrix ND at lines 5 and 6. Then, corresponding arc weights are calculated with ridge regression at line 7. At this point, note that MATLAB built-in function *ridge* can be used to calculate Ridge Regression. In the following loop, graph is filled with the explained rule set and the adjacency graph G is returned to the user.

3.3 BrainParcel

After constituting the neighborhood matrix ND and the edge weights of graph G, n-cut is used for clustering the graph. As explained in Chapter 2, n-cut is a graph partitioning algorithm which carries a graph cut method on a given undirected graph. Given graph G, n-cut cuts the edges one by one in an iterative manner. With each cut, graph is split into two smaller connected components. Readers can refer to Chapter 2 for detailed explanation of n-cut algorithm.

Letting N be the number of voxels, n-cut requires the representative graph G, which

Algorithm 3.4 Undirected Graph Construction from Mesh Model

```
1:  procedure GRAPH CONSTRUCTION
2:      N ← number of voxels
3:      G ← ZEROS(N, N)
4:      for i := 1 → N do
5:          indexes ← FIND NONZERO ENTRIES(ND(i, :))
6:          v_indexes ← v_j where j ∈ indexes
7:          arc_weights ← RIDGE(v_i, v_indexes)
8:          p ← LENGTH(indexes)
9:          for t := 1 → p_c do                    ▷ Make the graph undirected
10:             if G(i,t) == 0 then
11:                 G(i,t) ← arc_weights(t)
12:             else
13:                 if CORR(v_i, v_t) ≥ 0 then
14:                     G(i,t) ← MAX(G(i,t), arc_weights(t))
15:                 else
16:                     G(i,t) ← MIN(G(i,t), arc_weights(t))
        return G
```

is actually an $N - by - N$ adjacency matrix explained in the previous sections, and the number of intended brain parcels, $\mathbb{C}$. With graph cut operations, graph is split into C connected components where $C \leq \mathbb{C}$. Each sample is a member of one of this clusters and assigned with a cluster index. In other words, n-cut returns an $1 - by - N$ dimensional vector $\mathbb{L}_{\mathbb{C}} = [l_1^c, l_2^c, \ldots, l_N^c]$ where each l_i^c is a number between 1 and C. The n-cut segmentation method, as applied to undirected graph G is called BrainParcel. The output of this algorithm yields a set of supervoxels, which are homogeneous with respect to the subgraphs of mesh network.

Recall that, anatomical regions (AAL) produce a neuroscientific parcellation of the brain. In order to compare the brain decoding performances, we conducted our experiment, where we form mesh network both among anatomical regions and the network formed among supervoxels obtained at the output of BrainParcel. There are 116 brain regions in AAL and each voxel resides in one and only one region. Let us represent

the anatomically defined region indices of voxels with $\mathbb{L}_{\mathbb{A}} = [l_1^a, l_2^a, \ldots, l_N^a]$, in order to avoid confusions. Notice that, with $\mathbb{L}_{\mathbb{A}}$ we skip all of the brain parcellation steps. This method is widely used in the literature. Also, let us call $\mathbb{L} = [l_1, l_2, \ldots, l_N]$ to all kinds of brain segmentations; in our case it means $\mathbb{L} \supset (\mathbb{L}_{\mathbb{C}} \cap \mathbb{L}_{\mathbb{A}})$.

With the available supervoxel indexes, we need to calculate a representative signal for each supervoxel. For this purpose, we take an average among the time series of voxels within each supervoxel. With C supervoxels, we calculate set of vectors $U = [u_1, u_2, \ldots, u_C]$, where each u_i is the representative vector of supervoxel i and they are calculated as follows;

$$u_i = \frac{\sum_{l_j==i} v_j}{\sum_{l_j==i} 1}. \tag{3.5}$$

In the dataset on which we have carried our experiments, six measurements were taken for each cognitive stimulus. With this knowledge in mind, and assuming K stimuli were shown to subjects, time series of each voxel has a length $\mathbb{K} = 6K$. Therefore, as a result of the clustering we construct a data matrix U of size $C-by-\mathbb{K}$, where each row represents a feature, and each column is a cognitive stimulus.

3.4 Constructing Meshes for Voxels and Supervoxels - Estimation of MADs

The original area of utilization of the mesh model is to model the relationships among voxels and use this relationship for classification purposes. Both spatial and functional neighborhood were considered, and it has shown its powers in terms of classification accuracy. Functional Mesh Model (FMM) outperform most MVPA and Spatial Mesh Model (SMM) results. Therefore, we use FMM for classifying the cognitive states.

Data matrix U, that is acquired in the previous section, is feed into the FMM algorithm. Each supervoxel u_i is represented by linear combination of its neighbors, and this linear estimation is repeated for each cognitive stimuli. Recall that fMRI collects multiple measurements for each cognitive stimulus, in our dataset 6 measurements

are collected for each, and u_i is a vector of length $\mathbb{K} = 6K$ for K stimuli. Let us represent the length six vector of the stimulus k by $u_i(k)$. First, functionally closest p_m neighbors of $u_i(k)$; $\eta^{p_m}_{u_i(k)}$ are selected from the supervoxels u_j which has the highest correlation with voxel u_i according to the correlation function in Equation 3.3. Then, the mean square error $E(\epsilon^2_{i,j})$ is minimized to estimate $a_{i,j,k}$ of the following equation;

$$u_i(k) = \sum_{u_j(k) \in \eta^{p_m}_{u_i(k)}} a_{i,j,k} u_j(k) + \epsilon_{i,j}, \tag{3.6}$$

where $u_i(k)$ indicates the i^{th} voxel time series of a cognitive stimulus k, $u_j(k) \in \eta^{p_m}_{u_i(k)}$ shows the j^{th} neighboring supervoxel of $u_i(k)$ among total of p_m neighbors. Weights of this calculation $a_{i,j,k}$, are called Mesh Arc Descriptors (MADs) [20]. MADs are concatenated so that they represent a more powerful feature space compared to the raw fMRI signal intensity values that is used with MVPA studies. By constructing the meshes around each voxel/supervoxel, we concatenate the MADs and represent the stimuli in a feature space formed by MADs.

The following matrices shows the feature space at supervoxel level. Each $u_i(k)$ is a row vector with each scalar is the average intensity value obtained for that supervoxel. With mesh model, we calculate a column vector of length p_m, where each row is a feature.

$$u_i(k) = \begin{bmatrix} u_{i1} & u_{i2} & u_{i3} & u_{i4} & u_{i5} & u_{i6} \end{bmatrix} \rightarrow \begin{bmatrix} a_{i1k} \\ a_{i2k} \\ \vdots \\ a_{ip_mk} \end{bmatrix}$$

If we consider the whole time series of a supervoxel u_i, we represent each one with a matrix of size $p_m - by - K$ with K stimuli and p_m features.

$$u_i = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1\mathbb{K}} \end{bmatrix} \rightarrow \begin{bmatrix} a_{i11} & a_{i12} & a_{i13} & \dots & a_{i1K} \\ a_{i21} & a_{i22} & a_{i23} & \dots & a_{i2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{ip_m1} & a_{ip_m2} & a_{ip_m2} & \dots & a_{ip_mK} \end{bmatrix}$$

Extending this idea into the data matrix U, we change the representation into a matrix MAD_{sv} of size $(C - by - p_m)byK$ with K samples and $(C - by - p_m)$ features.

$$U = \begin{bmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1\mathbb{K}} \\ u_{21} & u_{22} & u_{23} & \dots & u_{2\mathbb{K}} \\ \dots & \dots & \dots & \dots & \dots \\ u_{C1} & u_{c2} & u_{C3} & \dots & u_{C\mathbb{K}} \end{bmatrix} \rightarrow \begin{bmatrix} a_{111} & a_{112} & a_{113} & \dots & a_{11K} \\ a_{121} & a_{122} & a_{123} & \dots & a_{12K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1p_m1} & a_{1p_m2} & a_{1p_m2} & \dots & a_{1p_mK} \\ a_{211} & a_{212} & a_{213} & \dots & a_{21K} \\ a_{221} & a_{222} & a_{223} & \dots & a_{22K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{2p_m1} & a_{2p_m2} & a_{2p_m2} & \dots & a_{2p_mK} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{C11} & a_{C12} & a_{C13} & \dots & a_{C1K} \\ a_{C21} & a_{C22} & a_{C23} & \dots & a_{C2K} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{Cp_m1} & a_{Cp_m2} & a_{Cp_m2} & \dots & a_{Cp_mK} \end{bmatrix} = MAD_{sv}$$

3.5 Classification

In the previous sections, we developed two brain networks of different sizes. First, we estimated a mesh network among the voxels. Then, by using the average time series of supervoxels, we estimated a coarser mesh network. Note that, in the first network, the number of nodes, which is equal to the number of voxels, are much larger than that of the second network, where the number of nodes is equal to the number of supervoxels. In order to measure improvements of our methods we have designed to experiment set ups. In the first design, we have used classification on the average time series of supervoxels, and in the second design we have experimented using the hierarchical architecture called Cognitive Learner (CL) that we propose in this thesis.

3.5.1 Brain Decoding by Supervoxels

In order to measure the representation power the supervoxels obtained by BrainParcel, we have carried classification using the MADs of supervoxels. MADs acquired from supervoxels are used to train a classifier. Logistic Regression is selected as a classifier for the network representations of anatomical regions, and supervoxels. Logistic Regression classifier produces the class conditional probabilities for each class. 6-fold cross validation schema is used for classification. Related information about the dataset and the selection of hyper parameters can be found in the next chapter.

3.5.2 Cognitive Learner

Arguing that using only the coarse representation of supervoxels may cause the loss of information, we propose a two layered architecture for cognitive state classification called Cognitive Learner (CL). General architecture of CL is given in Figure 3.3. As shown in the figure, voxel time series gathered from fMRI is the first level of the architecture. Second layer contains the average time series of the super voxels. These super voxels can be obtained by using anatomical regions or the output of BrainParcel algorithm. Therefore two representations of the stimuli of different granularity is

gathered. Then, a mesh network is formed at each level for voxels and supervoxels and MADs are extracted. For each mesh network an independent classifier is trained and class conditional probabilities are obtained. Finally, an ensemble learning algorithm, called Fuzzy Stacked Generalization, is used for fusing the output of the independent classifiers which classify the cognitive stimuli.

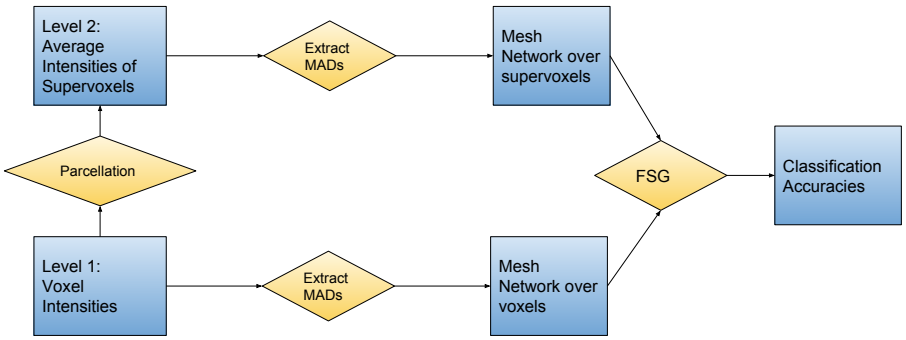


Figure 3.3: Detailed architecture of Cognitive Learner. First layer consists of the voxel intensity values and second layer is the average brain segments. At each layer relationships among voxels/super-voxels is computed. FSG is used to fuse information from the two layers.

3.5.2.1 Ensemble Learning: Fuzzy Stacked Generalization (FSG)

The Mesh Arc Descriptors (MADs) obtained at voxel and super-voxel level represent the stimuli with different granularities. In order to fuse the information, an ensemble learning algorithm called "Fuzzy Stacked Generalization" (FSG) is used [23]. FSG is a two-layer ensemble learning method. At base layer, mesh network obtained from voxel intensities and average super-voxel intensities resides, and two independent classifiers are trained to obtain class conditional membership values. Logistic regression is selected as the base layer classifier. Each classifier i returns a membership value μ_i. All membership values of the two classifiers are concatenated and form the input for meta-layer $\mathbf{M} = [\mu_1, \mu_2 \ldots \mu_L]$. Another logistic regression classifier is trained using $\mathbf{M}$.

Figure 3.4 shows the detailed architecture for FSG used in this study. MAD_{voxel} and MAD_{sv} represents the mesh networks obtained from voxel level and supervoxel

level respectively. For each, an independent logistic regression classifier is trained and gives the class conditional membership values M_1 and M_2. Then, M_1 and M_2 are concatenated to form **M**, and **M** is fed into another independent logistic regression classifier to classify the cognitive states.

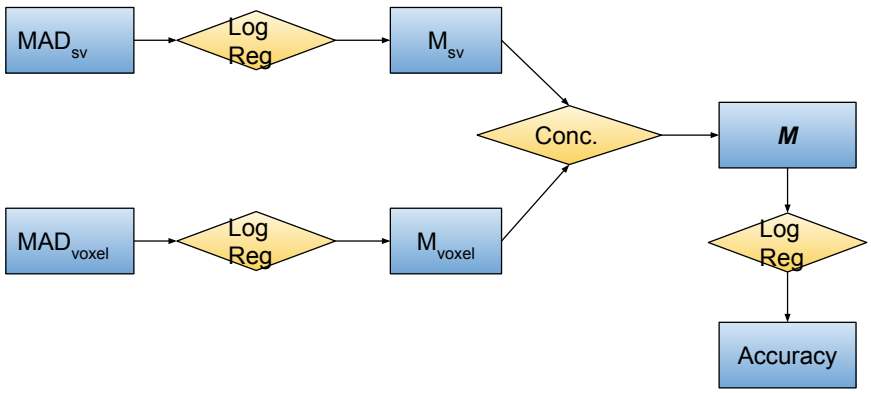


Figure 3.4: Detailed architecture of Fuzzy Stacked Generalization. Each base layer feature space is fed into a different classifier and resulting class conditional probabilities are used to form a new feature space to be classified with a meta layer classifier.

3.6 Chapter Conclusion

In this chapter, we have provided detailed information on our suggested segmentation and decoding methods, namely, BrainParcel and Cognitive Learner. First, the definitions of spatial and functional neighborhood are given. Then, forming the brain graph with mesh arc weights, based on the spatial and functional neighborhood systems is explained. Since we have used n-cut as the main parcellation algorithm and it requires a symmetrical graph, we needed a thresholding method to form an undirected brain graph. Brain graph is parcellated with n-cut to obtain voxel groups, called supervoxels. Then, we form a set of local meshes at voxel and supervoxel levels to represent a cognitive state at different network granularity. The arc weights of these two mesh networks are used as the input of two independent classifiers, which forms the Cognitive Learner. In the second part, we give detailed explanation of the Cognitive Learner architecture. Information acquired from the MADs of the voxel layer and supervoxel

layer are fused together with an ensemble learning architecture in order to increase the representational power.

CHAPTER 4

EXPERIMENTS ON BRAIN PARCELLATION, HIERARCHICAL NETWORK REPRESENTATION AND BRAIN DECODING

"The more we live without lies, the better"
"Ne kadar yalansız yaşarsak o kadar iyi"
-Can Yücel

This chapter gives the detailed experimental results on brain decoding in chronological order. First we introduce the dataset. Then we generate the supervoxels using the proposed BrainParcel algorithm. The power of BrainParcel is tested by classifying the cognitive states by the Mesh Network, estimated over the supervoxels. For comparison, counterparts of BrainParcel are also tested on our dataset, and those results are provided at each step. Since BrainParcel is highly sensitive to the parameter change, we repeated the experiments for multiple parameter sets and show the results for all. In the conclusion part, we compare the best results and emphasize the enhancement that we achieved. Moreover, for one parameter set, we show the results for the hierarchical Cognitive Learner architecture and show that Cognitive Learner with BrainParcel as a parcellation algorithm can achieve higher performances than the existing models.

4.1 Dataset

In this study we use a dataset called "Object Experiment". This dataset is recorded by the team of ImageLab METU members at Bilkent University UMRAM Center. It

consists of 4 subjects in the age of twenties. Each subject is shown various bird and flower pictures. In between those stimuli, simple mathematics questions are shown as transition. There are 6 runs in the experiment and in each run, 36 pictures are shown, which means, there are 216 samples in total. Number of samples are balanced for the two classes (bird and flower). Preprocessing of the dataset is carried with the toolbox SPM and the available number of voxels is around 20.000 for each subject. Also, there are 116 labeled anatomical regions, defined under MNI coordinate system [30].

Through the chapter, we provide experimental results, where each given accuracy is the output of a six fold cross validation. Recall that, each subject is given 6 runs of stimuli. At each fold, we split a run for testing and use the other 5 runs for training. Each given accuracy is the average of these 6 folds for each subject.

4.2 Performances of Brain Decoding by Mesh Network, Estimated Among Anatomical Regions

Anatomical regions constitute the baseline for all brain parcellation studies. Therefore, we have experimented classification with anatomical regions on our dataset. Within each of the 116 anatomical regions, we have computed the average signal intensity values of the voxels. Therefore, we obtain a new feature space of size 116, formed by these average signals to represent the cognitive stimuli. By using these raw signals we have computed classification accuracy with logistic regression. This approach is widely used in brain decoding literature and it is called Multi Voxel Pattern Analysis (MVPA). Result is shown on the first bar of Figure 4.1. Being 50.66%, MVPA has a chance level performance. In order to achieve higher levels of performance, we estimated a mesh network among the representative average signals of the anatomical regions and extracted MADs for classification as explained in Chapter 3. Rather than using raw signals, have used the MADs for classification. Number of neighbors of the mesh network, p_m, is tested within the set $[10, 25, 50, 75]$ and the regularization parameter λ of Equation 2.7 shown by l_m is tested in the set $[10, 25, 50, 75]$. Corresponding accuracies are also shown in Figure 4.1. As shown,

accuracies get higher with the use of MADs from mesh model. However, they are still closer to chance level. As an alternative approach, we suggest to use a task dependent brain parcellation rather than using neuroscientifically defined anatomical regions.

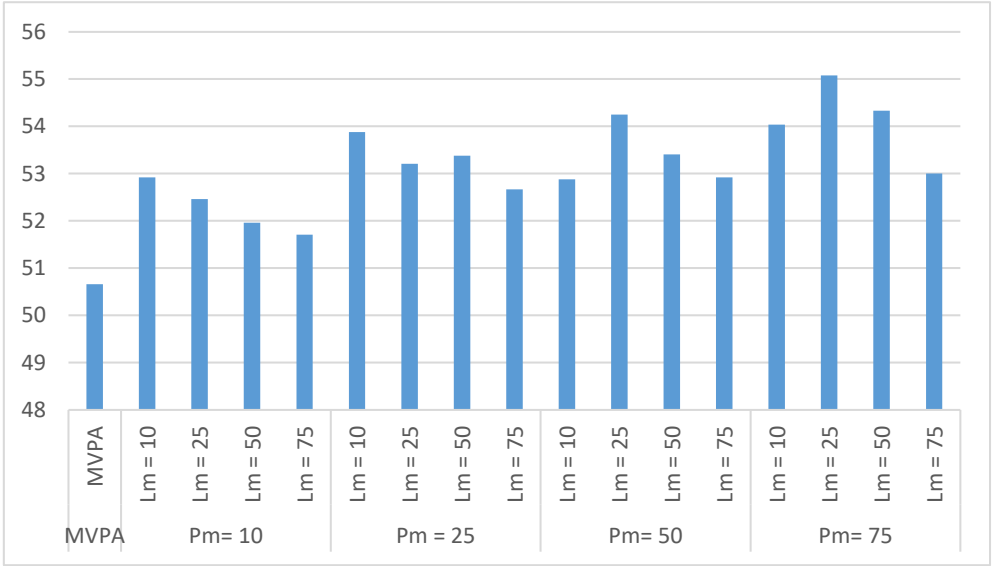


Figure 4.1: Accuracies obtained by the usage of anatomical regions. Leftmost bar shows the accuracy obtained by classifying the raw signals. The rest of the bars show the classification accuracies of mesh model for different neighborhood and regularization parameters.

4.3 Performances for Brain Decoding with Classical N-cut Segmentation

As explained in Chapter 2, spatially constrained n-cut is used successfully for brain segmentation as a substitution of anatomical regions by Craddock et. al. [6]. Since this study stems from this idea, we conducted a similar approach to the ones in the literature on our dataset.

We formed a brain graph, where each voxel is connected to its 26 spatial neighbors

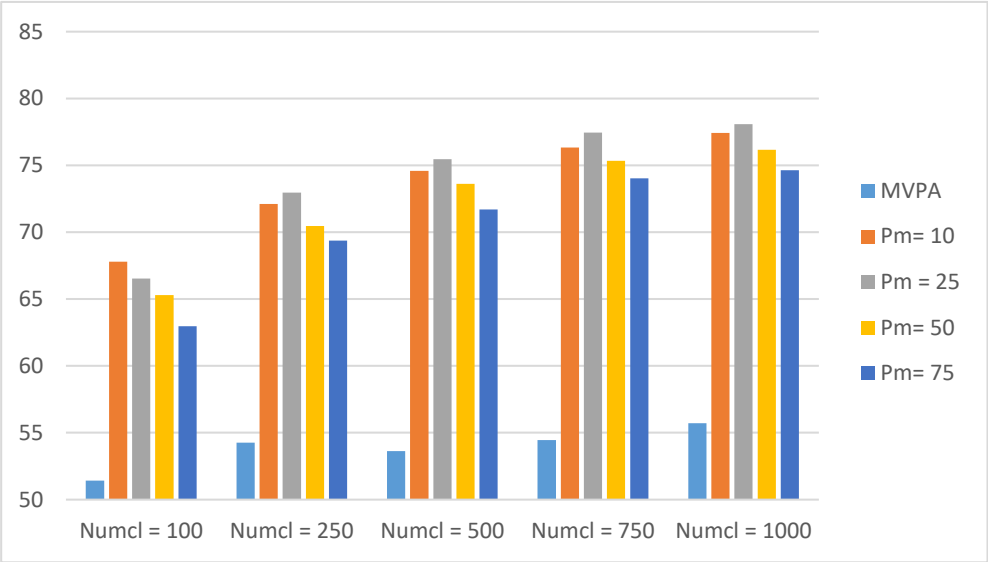


Figure 4.2: Classification accuracies of classical n-cut, showing that mesh model with the regularization parameter, $l_m = 10$ gives highest accuracies regardless of the number of supervoxels and number of neighbors, p_m

and the edge values are the calculated by the correlation between the intensities of two voxels. Unlike Craddock et. al. we did not put any constraints on the edge values to be greater than 0.5. Then, we partitioned this graph and obtain the brain parcels with n-cut. Within each parcel we computed the average time series of the voxels. Note that, we have obtained different number of brain parcels and these average time series form a new feature space for the cognitive stimulus. By using the raw average time signals, we computed classification accuracies. Bars labeled with MVPA on Figure 4.2 gives the accuracies of this experiment.

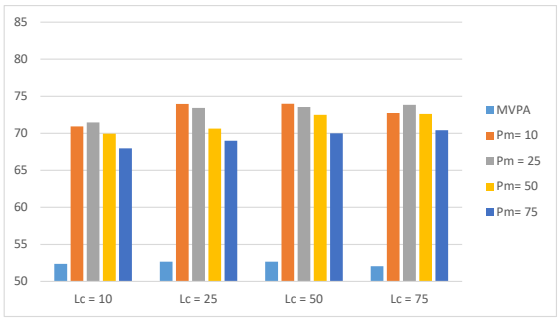
As shown in Figure 4.2, the classification accuracies of MVPA are at almost chance level. Therefore, we used mesh model for classification. We have constructed a mesh network among brain parcels and extracted MADs. This experiments tests the number of neighbors $p_m = [10, 25, 50, 75]$, and regularization parameter, $l_m =$

$[10, 25, 50, 75]$. Figure 4.2 shows the accuracies for different parcellation parameters. Recall that we have experimented different number of supervoxels; which is represented by NumCl such that; $NumCl = [100, 250, 500, 750, 1000]$. Each group on Figure 4.2 gives classification accuracies obtained with a different number of brain parcels. This figure also points out that regardless of the number of supervoxels, $NumCl$, and p_m, mesh model network achieves higher accuracies when λ parameter of the estimation $l_m = 10$. As shown, accuracy increases as the number of segments increase and we can achieve significantly higher accuracies with the use of mesh model. Moreover, comparing this figure with Figure 4.1 of anatomical regions, one can detect a significant raise in the classification accuracies of cognitive states.

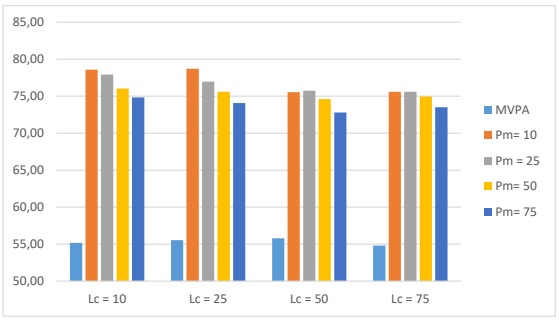
4.4 Performances of Brain Decoding with Spatially Constrained BrainParcel

Considering the improvements achieved by mesh model in the previous studies in terms of classification, we build a brain parcellation model BrainParcel. In Spatial BrainParcel, we form a mesh network around each voxel, where neighborhood is defined spatially with $Spatial_ND$ to form a brain graph. Number of neighbors at parcellation step p_c is selected to be 26, to be comparable with existing studies, and λ parameter of the ridge regression l_c is tested within the set $[10, 25, 50, 75]$. Notice the difference in the meanings between p_c and p_m, and between l_c and l_m. Subscript c represents the mesh network parameters at the parcellation step while we construct the arc weights of the brain graph over mesh model, while subscript m represents the same parameters at classification step where we extract MADs over mesh model.

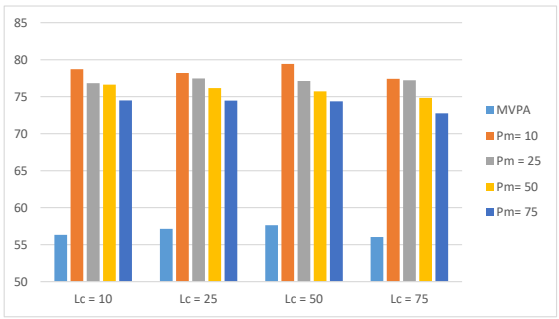
After constructing the graph, brain parcellation is obtained by n-cut algorithm to obtain supervoxels. Number of supervoxels is varied in the range $[100, 250, 500, 750, 1000]$. Then, within each supervoxel, representative average time series is computed and a mesh network is formed among these representative signals to extract MADs. As in the previous experiments, p_m and l_m for the MADs are tested from the set $[10, 25, 50, 75]$.



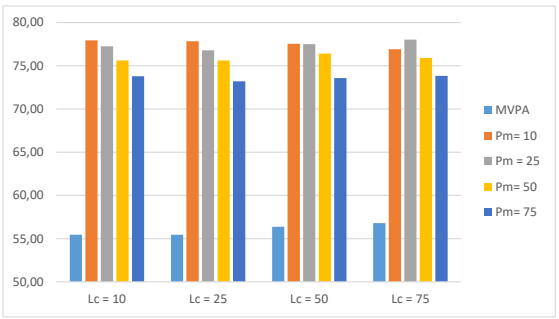
(a) Number of supervoxels = 100



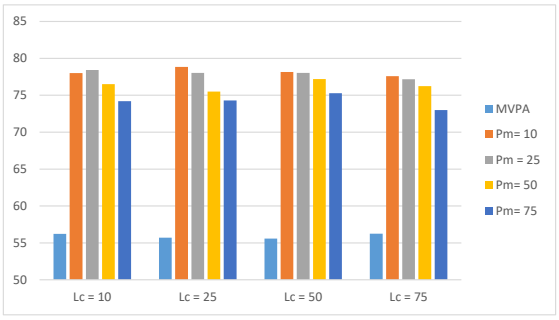
(b) Number of supervoxels = 250



(c) Number of supervoxels = 500



(d) Number of supervoxels = 750



(e) Number of supervoxels = 1000

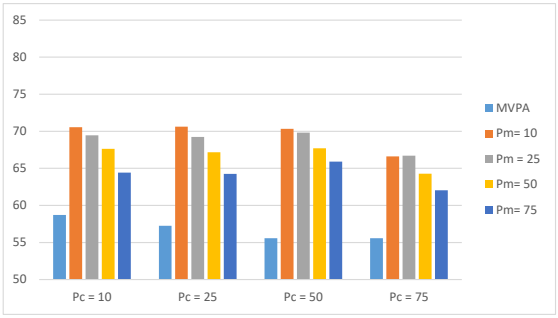
Figure 4.3: Classification accuracies of spatial BrainParcel, showing that mesh model with $l_m = 10$ gives highest accuracies, regardless of the number of supervoxels and p_m.

Figure 4.3 shows the classification accuracies obtained by different p_m and l_m parameters. Each graph shows the accuracies for different number of supervoxels. Notice that, we get highest accuracies with $l_m = 10$. Each figure shows experimental results for different number of supervoxels. Each group in each figure represents a different l_c parameter at parcellation step where p_c is 26. Notice that, leftmost bars are the MVPA accuracies obtained by raw signal time series. Moreover, performance gets higher with increasing number of supervoxels. When Figure 4.2 and 4.3 is compared, one can see an increase over classification accuracies with the usage of Spatial BrainPacel.

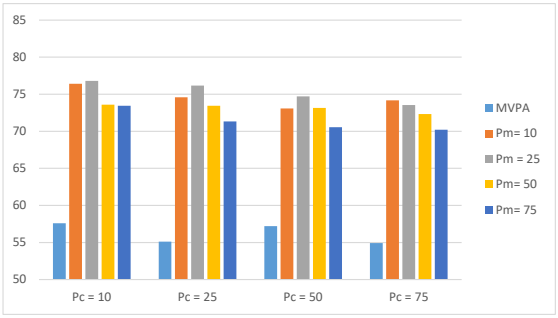
4.5 Performances for Brain Decoding of Classical N-cut with Functional Constraints

Recall that, computing the mesh networks among functional neighbors can improve the classification accuracies [19, 20]. Observing these improvements, we forced similar functional constrains on brain parcellation. While $Spatial_ND$ forms a brain graph by linking each voxel by its spatial neighbors, we use $Functional_ND$ where each voxel is connected to its functional neighbors in the brain graph. Rest of the procedure is exactly the same with the previous experiments. We form a brain graph with $Functional_ND$, where edge values of the graph are the correlation values between the intensities of two voxels. Then we partition this graph into user defined number of brain parcels. Similarly, number of brain parcels $NumCl$ are varied within the set $100, 250, 500, 750$ and 1000. Then, average time intensity values are computed within each brain parcel, and a mesh network are formed among those signals to extract MADs. Other parameters spaces are as follows; $p_c = [10, 25, 50, 75]$, $p_m = [10, 25, 50, 75]$, $l_m = [10, 25, 50, 75]$. Notice that, since we construct the graph by correlations, we do not use l_c of mesh model in this experiment.

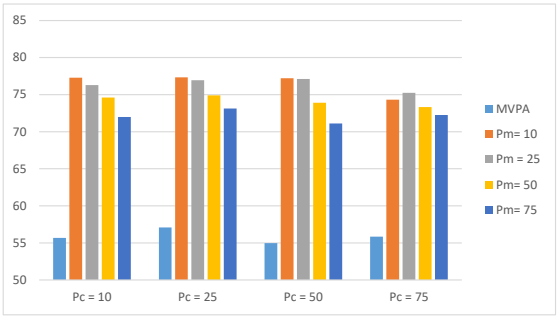
Figure 4.4 shows the classification accuracies of this experiment. Each plot shows the accuracies for a different number of brain parcels. Left-most bar in each graph gives the MVPA accuracies obtained by classifying the raw intensity values of the brain



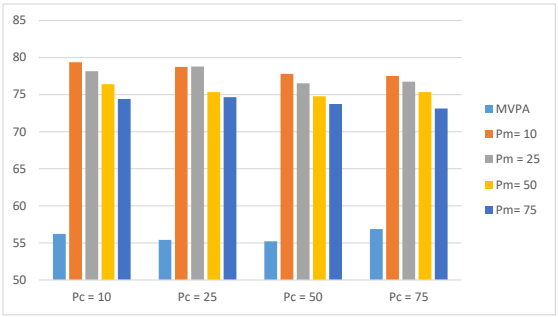
(a) Number of supervoxels = 100



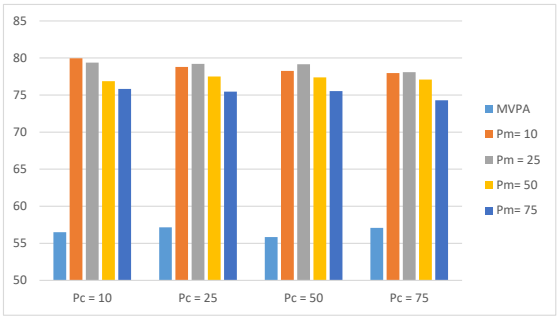
(b) Number of supervoxels = 250



(c) Number of supervoxels = 500



(d) Number of supervoxels = 750



(e) Number of supervoxels = 1000

Figure 4.4: Classification accuracies of classical n-cut with functional constrained neighborhood, showing that mesh model with $l_m = 10$ gives highest accuracies regardless of the number of brain parcels and p_m.

parcels. Other bars gives the accuracies obtained using the MADs. As previously noted, $l_m = 10$ gives higher accuracies regardless of other parameters. These results show a similar results about the number of brain parcels, such that highest accuracies are achieved with the most available number of brain parcels.

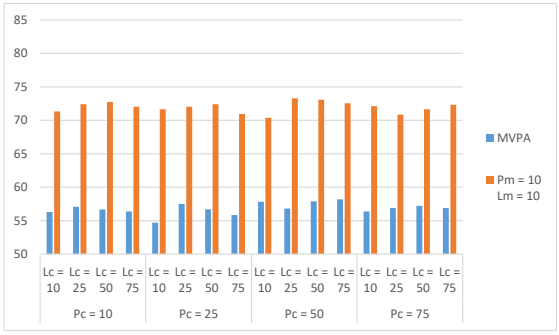
4.6 Performances of Brain Decoding with Functional Constrained BrainParcel

As the last part of experiments performed on BrainParcel, we build Functional BrainParcel. While the neighborhood is defined by $Functional_ND$, graph construction is carried by the rules of BrainParcel. Procedure is completed with the set of parameters $p_c = [10, 25, 50, 75]$, $l_c = [10, 25, 50, 75]$, $p_m = [10, 25, 50, 75]$, $l_m = [10, 25, 50, 75]$. Figure 4.5 indicates that we can get higher accuracies with $l_m = 10$ as reported in the previous experiments.

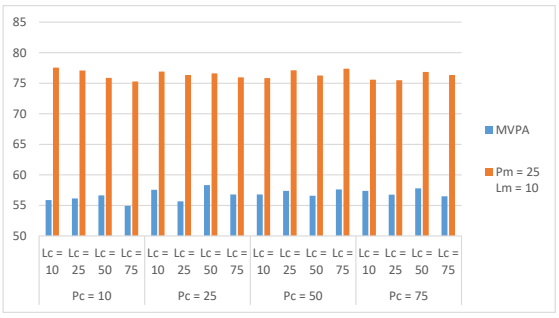
Figure 4.5 shows the accuracies for this experiment setup. Each plot shows the accuracies for a different number of brain parcels. Left-most bar in each graph gives the MVPA accuracies obtained by classifying the raw intensity values of the average supervoxels. Figure 4.5 indicates that we can get higher accuracies with $l_m = 10$ as previous experiments indicated.

4.7 Comparative Results of BrainParcel

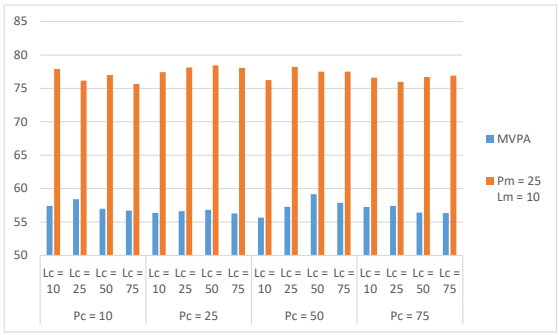
In this section, we provide a comparison between the brain parcellation algorithms that has been covered so far. Table 4.1 shows the overall results. Note that, these are the best results that has been achieved with the optimal parameters. Recall that, functionally constrained systems that construct the graph with $Functional_ND$ neighborhood system does not provide any spatial integrity within the brain parcels, since the brain graph is not formed on these grounds. On the other hand, spatially constrained systems provide both spatial continuity and functional homogeneity since the brain graph is formed by spatial restrictions and edges are weighted in terms of functional connectivity.



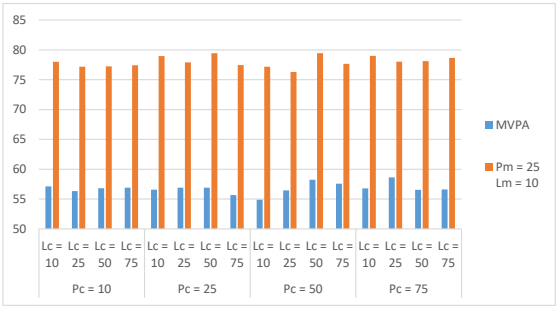
(a) Number of supervoxels = 100



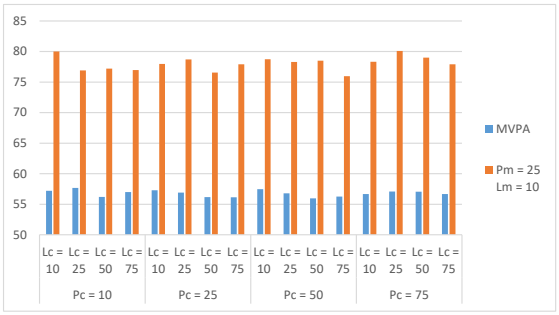
(b) Number of supervoxels = 250



(c) Number of supervoxels = 500



(d) Number of supervoxels = 750



(e) Number of supervoxels = 1000

Figure 4.5: Classification accuracies of Functional BrainParcel, showing that mesh model with $l_m = 10$ gives highest accuracies regardless of the number of supervoxels and p_m.

In Table 4.1, the first and third column give the best results for the classical approaches, and the other two gives the results obtained by BrainParcel that we have offered. Each row of this table gives the results for a different number of supervoxels. Notice that spatially constrained BrainParcel gives the best results in the overall schema.

These results point to the idea that, in order to achieve better representational power in terms of cognitive state classification, one needs spatially contiguous and functionally homogeneous brain parcels, which is accomplished by spatial BrainParcel. Moreover, recall that we have offered BrainParcel as an alternative to anatomical regions, which has 116 brain regions. When the results acquired with a closer number of supervoxels is considered, Spatial BrainParcel results in higher classification accuracies.

Table 4.1: Overall 2-class classification results acquired from the MADs of supervoxels. These results suggest that Spatial BrainParcel gives higher performances on average since it provides spatial continuity and functional homogeneity within brain parcels. These are the best results for all experiment setups.

	Spatial Constraints		Functional Constraints	
Number of segments	Classical	BrainParcel	Classical	BrainParcel
100	67.79	**74.00**	70.63	73.29
250	72.96	**78.71**	76.79	77.54
500	75.46	**79.42**	77.33	77.54
750	77.46	78.04	**79.38**	77.54
1000	78.08	78.83	79.96	**80.08**

4.8 Comparative results with Cognitive Learner

This section provides the experiments to measure the brain decoding accuracies obtained by the hierarchical learning architecture Cognitive Learner (CL). Note that all of the experiments are carried with $l_c = 10$, $p_m = 10$ and $l_m = 10$ for the sake of complexity. Also, number of neighbors at parcellation level, p_c, is chosen to be 26

for spatially constrained systems, and 10 for functionally constrained systems. First of all, consider Table 4.2, which shows the accuracies obtained at the parcellation level with these parameters. Table layout is the same with the previous one. Where each row shows the accuracies for a different number of supervoxels. Table 4.3 gives the accuracies obtained by fusing the information from the network of supervoxels from Table 4.2 and the network of the voxel level. As shown, proposed architecture increases the classification accuracies. Especially, for spatially constrained systems, we show a significant amount of increase in the brain decoding accuracies.

Table 4.2: 2-class classification accuracies obtained from the network over supervoxels with $l_c = 10$, $p_m = 10$ and $l_m = 10$.

	Spatial Constraints		Functional Constraints	
Number of segments	Classical	BrainParcel	Classical	BrainParcel
100	67.79	**70.92**	70.54	70.04
250	72.13	**78.5**8	76.42	76.79
500	74.58	**78.71**	77.29	78.00
750	76.33	77.96	**79.38**	76.42
1000	77.418	78.00	**79.96**	78.92

Table 4.3: 2-class classification accuracies obtained from the Cognitive Learner with $l_c = 10$, $p_m = 10$ and $l_m = 10$.

	Spatial Constraints		Functional Constraints	
Number of segments	Classical	BrainParcel	Classical	BrainParcel
100	75.69	76.16	**76.27**	75.34
250	77.08	**80.32**	77.77	78.47
500	78.58	**79.62**	78.24	78.24
750	77.54	**80.09**	78.81	78.70
1000	79.51	79.05	**79.74**	79.17

4.9 Chapter Conclusion

This chapter covers the experimental results obtained by the proposed methods and their counterparts, reported in the literature. Range of a parameter set is performed, and accuracies are reported for all combinations. By the results we can conclude that both spatial and functional homogeneity are important in brain segmentation studies. These characteristics can be achieved by both classical methods offered in the literature and Spatial BrainParcel that we propose.

Second part of this study includes the experiments carried on the architecture Cognitive Learner. For this setup, experiments are repeated for one parameter set which is shown to be optimal. Results show that Cognitive Learner with Spatial BrainParcel at base layer parcellation method provides better representation power and yields higher brain decoding performances compared to the state of the art brain parcellation and decoding methods.

CHAPTER 5

CONCLUSION AND FUTURE WORK

"Just before I go, I have things to do"
"Giderayak işlerim var bitirilecek, giderayak"
-Nazım Hikmet

Brain decoding studies take up an important place in the purpose of mind reading. fMRI datasets are used widely in this regard. In this study, we propose a hierarchical network architecture for brain decoding with fMRI dataset. This chapter gives the discussion on the two proposed methods, namely, BrainParcel and Cognitive Learner. We also suggest possible future direction in the light of the obtained results for the sake of progress in the scientific field on human cognition.

5.1 Discussion on BrainParcel and Possible Future Directions

In this study, we offer a spatial and functional brain parcellation methodology, which is presented as an alternative to the current segmentation studies in the literature [14, 29]. BrainParcel uses spectral clustering in its underlying structure, which represents the voxel space as a graph formed by mesh model. Common studies compute the edges of the graph as the correlation between two voxels, whereas we computed the edges by measuring the dependencies by mesh model among a group of voxels. Then, brain graph is partitioned with n-cut and supervoxels are obtained.

Experimental results verify the common view on brain parcellation studies in many

ways. As suggested, using task dependent methods can achieve better performances than anatomical regions. Moreover, in terms of characteristics of brain parcels, it is demonstrated that aiming for functional homogeneity, united with the spatial contiguity is the best approach.

Also, using mesh model for classification is a better approach in terms of classification accuracy than using the raw intensity values of the voxels. Results has shown that using the MADs of the mesh model for classification, results in significantly higher accuracies in all of the experiment setups.

Our study revealed that using mesh model not only for classification, but for creating the brain graph is a better approach since it has increased the classification accuracies regarding to other similar brain parcellation approaches. This increment is in small quantities in many cases, however when the size of the data set is considered, it is acceptable not to expect large impacts on classification with the innovations on brain parcellation schema.

Possible future directions regarding to the proposed brain parcellation can be listed as follows;

- Since we have shown that higher number of supervoxels provide higher accuracies, we need to repeat the trials with higher number of supervoxels, also parameter optimization should be performed with a wider hyperparameter space.
- Even if we do not expect much effects on classification accuracies, validation on the available brain parcels should be performed to compare the BrainParcel with others approaches.
- In order to analyze the effectiveness of the proposed algorithm, detailed statistical tests should be carried for BrainParcel in both functional and spatial restricted cases. Also, these results should be compared to the classical approaches, such as k-means [9, 17] or hierarchical clustering [2]
- Experiments should be repeated on a larger dataset, with higher number of subjects and cognitive stimuli, to verify the power of the BrainParcel.

5.2 Discussion on Cognitive Learner and Possible Future Directions

We argue that selecting only the voxel level or only the supervoxel level representation of fMRI data is not enough for cognitive state classification. Therefore, we propose a two level hierarchy for cognitive state classification of the cognitive states. Our experimental results confirm this opinion. Even with a non optimized parameter set, Cognitive Learner increases the classification accuracies for a two class classification problem.

Possible future directions regarding to the hierarchical architecture that we can offer be listed as follows;

- Experiments should be repeated for an optimized hyperparameter set.
- Since we have shown using two levels of different granularity increases the classification accuracies, we need to repeat experiments with more number of brain parcellation levels, with different granularities.
- Experiments should be repeated on a larger dataset with higher number of subjects and cognitive stimuli.

REFERENCES

[1] A. Afrasiyabi and F. T. Yarman-Vural. fmrg verilerinde temel bileşenler analizi ve özyinemeli boyut eliminasyonu kullanarak boyut küçültme. In *23th Signal Processing and Communications Applications Conference (SIU)*, pages 2438–2441. IEEE, 2015.

[2] S. Alkan and F. T. Yarman-Vural. Ensembling brain regions for brain decoding. In *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2948–2951. IEEE, 2015.

[3] T. Blumensath, S. Jbabdi, M. F. Glasser, D. C. Van Essen, K. Ugurbil, T. E. Behrens, and S. M. Smith. Spatially constrained hierarchical parcellation of the brain with resting-state fmri. *Neuroimage*, 76:313–324, 2013.

[4] D. Cordes, V. Haughton, J. D. Carew, K. Arfanakis, and K. Maravilla. Hierarchical clustering to measure connectivity in fmri resting-state data. *Magnetic resonance imaging*, 20(4):305–317, 2002.

[5] M. N. Coutanche, S. L. Thompson-Schill, and R. T. Schultz. Multi-voxel pattern analysis of fmri data predicts clinical symptom severity. *Neuroimage*, 57(1):113–123, 2011.

[6] R. C. Craddock, G. A. James, P. E. Holtzheimer, X. P. Hu, and H. S. Mayberg. A whole brain fmri atlas generated via spatially constrained spectral clustering. *Human brain mapping*, 33(8):1914–1928, 2012.

[7] M. C. Diamond, A. B. Scheibel, G. M. Murphy, and T. Harvey. On the brain of a scientist: Albert einstein. *Experimental neurology*, 88(1):198–204, 1985.

[8] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. John Wiley & Sons, 2012.

[9] G. Flandin, F. Kherif, X. Pennec, G. Malandain, N. Ayache, and J.-B. Poline. Improved detection sensitivity in functional mri data using a brain parcelling technique. *Medical Image Computing and Computer-Assisted Intervention, MICCAI*, pages 467–474, 2002.

[10] G. Flandin, F. Kherif, X. Pennec, D. Riviere, N. Ayache, and J.-B. Poline. Parcellation of brain images with anatomical and functional constraints for fmri data analysis. pages 907–910, 2002.

[11] H. Kober, L. F. Barrett, J. Joseph, E. Bliss-Moreau, K. Lindquist, and T. D. Wager. Functional grouping and cortical–subcortical interactions in emotion: a meta-analysis of neuroimaging studies. *Neuroimage*, 42(2):998–1031, 2008.

[12] L. I. Kuncheva and J. J. Rodríguez. Classifier ensembles for fmri data analysis: an experiment. *Magnetic resonance imaging*, 28(4):583–593, 2010.

[13] L. I. Kuncheva, J. J. Rodríguez, C. O. Plumpton, D. E. Linden, and S. J. Johnston. Random subspace ensembles for fmri classification. *IEEE transactions on medical imaging*, 29(2):531–542, 2010.

[14] T. W. Liao. Clustering of time series data—a survey. *Pattern recognition*, 38(11):1857–1874, 2005.

[15] M. Martínez-Ramón, V. Koltchinskii, G. L. Heileman, and S. Posse. fmri pattern classification using neuroanatomically constrained boosting. *Neuroimage*, 31(3):1129–1141, 2006.

[16] T. M. Mitchell, S. V. Shinkareva, A. Carlson, K.-M. Chang, V. L. Malave, R. A. Mason, and M. A. Just. Predicting human brain activity associated with the meanings of nouns. *science*, 320(5880):1191–1195, 2008.

[17] H. Moğultay, S. Alkan, and F. T. Yarman-Vural. Classification of fmri data by using clustering. In *23th Signal Processing and Communications Applications Conference, SIU*, pages 2381–2383. IEEE, 2015.

[18] K. A. Norman, S. M. Polyn, G. J. Detre, and J. V. Haxby. Beyond mind-

reading: multi-voxel pattern analysis of fmri data. *Trends in cognitive sciences*, 10(9):424–430, 2006.

[19] I. Onal, M. Ozay, E. Mizrak, I. Oztekin, and F. Yarman-Vural. A new representation of fmri signal by a set of local meshes for brain decoding. *IEEE Transactions on Signal and Information Processing over Networks*, 2017.

[20] I. Onal, M. Ozay, and F. T. Y. Vural. Functional mesh model with temporal measurements for brain decoding. In *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2624–2628. IEEE, 2015.

[21] I. Onal, M. Ozay, and F. T. Y. Vural. Modeling voxel connectivity for brain decoding. In *International Workshop on Pattern Recognition in NeuroImaging (PRNI)*, pages 5–8. IEEE, 2015.

[22] M. Ozay, I. Öztekin, U. Öztekin, and F. T. Y. Vural. Mesh learning for classifying cognitive processes. *arXiv preprint arXiv:1205.2382*, 2012.

[23] M. Ozay and F. T. Y. Vural. A new fuzzy stacked generalization technique and analysis of its performance. *arXiv preprint arXiv:1204.0171*, 2012.

[24] L. Pauling and C. D. Coryell. The magnetic properties and structure of hemoglobin, oxyhemoglobin and carbonmonoxyhemoglobin. *Proceedings of the National Academy of Sciences*, 22(4):210–216, 1936.

[25] W. D. Penny, K. J. Friston, J. T. Ashburner, S. J. Kiebel, and T. E. Nichols. *Statistical parametric mapping: the analysis of functional brain images*. Academic press, 2011.

[26] J. Richiardi, H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville. Decoding brain states from fmri connectivity graphs. *Neuroimage*, 56(2):616–626, 2011.

[27] X. Shen, X. Papademetris, and R. T. Constable. Graph-theory based parcellation of functional subunits in the brain from resting-state fmri data. *Neuroimage*, 50(3):1027–1035, 2010.

[28] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

[29] B. Thirion, G. Varoquaux, E. Dohmatob, and J.-B. Poline. Which fmri clustering gives good brain parcellations? *Frontiers in neuroscience*, 8, 2014.

[30] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Etard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated anatomical labeling of activations in spm using a macroscopic anatomical parcellation of the mni mri single-subject brain. *Neuroimage*, 15(1):273–289, 2002.

[31] K. Uludag, D. J. Dubowitz, and R. B. Buxton. Basic principles of functional mri. *Clinical MRI. Elsevier, San Diego*, pages 249–287, 2005.

[32] M. Van Den Heuvel, R. Mandl, and H. H. Pol. Normalized cut group clustering of resting-state fmri data. *PloS one*, 3(4):e2001, 2008.

[33] J. Wang and H. Wang. A supervoxel-based method for groupwise whole brain parcellation with resting-state fmri data. *Frontiers in human neuroscience*, 10, 2016.