

**PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
FACULTY OF INFORMATICS
GRADUATE PROGRAM IN COMPUTER SCIENCE**

**FEATURE SELECTION FOR
NEUROIMAGING APPLIED TO
WORD-CATEGORY
IDENTIFICATION IN DYSLEXIC
CHILDREN**

CAROLINE SELIGMAN FROEHLICH

Thesis presented as partial requirement for
obtaining the degree of Master in Computer
Science at Pontifical Catholic University of
Rio Grande do Sul.

Advisor: Prof. Felipe Rech Meneguzzi

**Porto Alegre
2015**

Dados Internacionais de Catalogação na Publicação (CIP)

F925f Froehlich, Caroline Seligman

Feature selection for neuroimaging applied to word-category
identification in dyslexic children / Caroline Seligman

Froehlich. – Porto Alegre, 2015.

84 f.

Dissertação (Mestrado) – Faculdade de Informática,
PUCRS.

Orientador: Prof. Dr. Felipe Meneguzzi.

1. Informática. 2. Diagnóstico por Imagem.
3. Processamento de Imagens. 4. Imagem por Ressonância
Magnética. 5. Dislexia. I. Meneguzzi, Felipe. II. Título.

CDD 006.61

**Ficha Catalográfica elaborada pelo
Setor de Tratamento da Informação da BC-PUCRS**



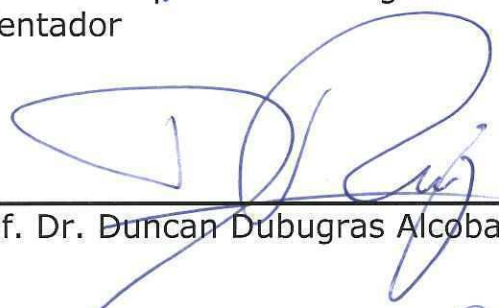
Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "*Feature Selection for Neuroimaging Applied to Word-Category Identification in Dyslexic Children*" apresentada por Caroline Seligman Froehlich como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, aprovada em 26/02/2015 pela Comissão Examinadora:



Prof. Dr. Felipe Rech Meneguzzi – PPGCC/PUCRS
Orientador



Prof. Dr. Duncan Dubugras Alcoba Ruiz – PPGCC/PUCRS



Prof. Dr. Alexandre Rosa Franco – FENG/PUCRS



Prof. Dr. Dante Augusto Couto Barone – UFRGS

Homologada em...../...../....., conforme Ata No. pela Comissão Coordenadora.

Prof. Dr. Luiz Gustavo Leão Fernandes
Coordenador.

PUCRS

Campus Central

Av. Ipiranga, 6681 – P32– sala 507 – CEP: 90619-900
Fone: (51) 3320-3611 – Fax (51) 3320-3621
E-mail: ppgcc@pucrs.br
www.pucrs.br/facin/pos

LIST OF FIGURES

- Figure 2.1 – The 4 lobes of the brain. The configuration of the lobes in both hemispheres are replicated. The frontal lobe (green) is related with planning functions. The temporal lobe (red) is related to language functions and giving meaning to visual inputs. The parietal lobe (blue) is related to processing sensory inputs and contains the motor areas. The occipital lobe (gray) is related to visual processing. The two important language-related areas are the Broca’s area (or inferior frontal gyrus) in yellow, related to speech production and the Wernicke’s area (or superior temporal gyrus) in purple related to speech processing. 23
- Figure 2.2 – Differences between brain activation when children are resting and performing the pseudo-word task. The orange areas indicate where there is ‘ activation in the pseudo-word task than in rest across subjects. 25
- Figure 2.3 – Differences in the blood flow when a brain region is in rest and activated. In rest condition, we have the same level of oxy and deoxyhemoglobin. Conversely, when there is neural activity, the blood flow increases along with the oxyhemoglobin level. That happens in order to supply local oxygen demands. As the oxyhemoglobin level increases, the deoxyhemoglobin level decreases, which makes the local magnetic resonance signal increases too. Inspired in [Maz09] 26
- Figure 2.4 – After brain activity, there is immediate oxygen consumption; 2 seconds later, more oxyhemoglobin arrives in the activated area in order to supply the oxygen demand. As there is more oxygen (and oxyhemoglobin) than the required area needs, the local magnetic resonance signal increases. The signal reaches the maximum value at 5 seconds, and subsequently returns to the baseline value at 12 seconds. The BOLD signal changes only 3% when a brain area is activated. 26
- Figure 2.5 – Schematic example of 2 different paradigm blocks using the pseudo-word task. In the block design paradigm, just one condition is presented in a single block, while in event related all conditions are presented randomly in a single block. All trials are followed by a few seconds of rest. 28
- Figure 2.6 – ICA algorithm applied in rsfMRI data, generating 20 components. The image is generated with the scikit-learn [PVG⁺11] implementation of ICA 29
- Figure 3.1 – $fv1$ and $fv2$ graphic, where x axis is the frequency of the word *test* and the y axis is the frequency of the word *server* 35
- Figure 3.2 – Estimating the class of the new example (gray) by calculating its distance from the other examples 40
- Figure 3.3 – Data linearly separable, with the support vectors inside circles. The separating lines are under the support vectors showing the boundaries between the two classes. 41

Figure 3.4 – Non-linear separable data, separated by the black line	41
Figure 4.1 – Example of a Region of Interest. This ROI covers all voxels from the inferior frontal part of the brain, which is involved in the language production task. Thus, it is appropriate for neuroimaging studies about reading, as ACERTA project. . . .	46
Figure 4.2 – 3 types of parcellations: AAL divides the brain in 116 anatomical regions, cc200 divides the brain in 200 functional regions and cc400 divides the brain in 400 functional regions. From Craddock et al. [CJH ⁺ 12]	48
Figure 4.3 – The shape of the cluster ReHo algorithm analyzes to find clusters with high similarity. Each cube is a voxel. from C-PAC [SCK ⁺ 14]	49
Figure 4.4 – Binary mask used for extracting all the voxels that are inside the brain. . . .	50
Figure 5.1 – From [CHHM09]. Connectivity matrix showing how 15 brain regions communicate. The connectivity matrix varies according to the feature selection method. Yellow squares show positive correlation between two areas while blue squares show negative correlation. Gray squares show no correlation between areas.	52
Figure 5.2 – From [PTP ⁺ 01]. Brain areas activated during the print word stimulus that correlate with writing behavioral measures. That means poor readers activate less the highlighted areas than good readers. The left hemisphere is at the right side of the image.	57
Figure 6.1 – Schematic view of experiments from data acquisition through identifying the important brain regions for reading. First, in the scanning session, children read a list of words while the MRI machine acquire the raw fMRI data. Second, the raw data is processed, cleaning the noisy fMRI data and generating contrast images. Third, the contrast image is used in the feature selection algorithm. In this example, we use the whole brain feature selection, in which all voxels in yellow (that are inside the brain) are chosen. The feature selection algorithm generates examples for classification. Finally, we use the examples with the SVM classifier, which show us the important regions for reading.	59
Figure 6.2 – Two tasks of different conditions can generate different BOLD signals (tasks A and B). Task B starts right after task A, but the BOLD signal of task A do not have en ought time to return to baseline before task B starts. When the tasks are too close in time, the resulting BOLD signal of the two tasks sum (Task A + B). .	62
Figure 6.3 – Schematic example of contrast images <i>all > baseline</i> and <i>baseline > all</i> generation. First, we calculate the average activation in conditions <i>all</i> and in condition <i>baseline</i> . In this example, in the left figures, there is more activation in the back of the brain in <i>all</i> and more activation in the right hemisphere in <i>baseline</i> . Second, we subtract one condition from another, generating the contrast images in the right. For generating <i>all > baseline</i> image, we subtract <i>all</i> from <i>baseline</i> . Conversely, for generating <i>baseline > all</i> image, we subtract <i>baseline</i> from <i>all</i> . . .	63

Figure 6.4 – Location of the 10 parcellations that contribute the most to classification from the cc200 parcellations. 70

Figure 6.5 – Brain regions that contribute the most for classification using voxels from the whole brain. The regions are the 5% most important voxels that belongs to clusters of at least 100 voxels. Top left: All x Baseline classification; Top right: Regular x Irregular classification; Bottom left: Regular x Pseudo classification; Bottom Right: Irregular x Pseudo classification. 71

Figure 6.6 – Average classification accuracy from the 4 classifiers defined in Table 6.2. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are left and right precentral and lingual regions and right postcentral region. 72

Figure 6.7 – Accuracy from all x baseline classifier. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are left and right occipital and parietal areas, left insula and inferior frontal operculum. 73

Figure 6.8 – Accuracy from regular x irregular classifier. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are left and right occipital areas, left pallidum and insula and right frontal region. 74

Figure 6.9 – Accuracy from regular x pseudo classifier. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are right inferior frontal, parahippocampal and anterior cingulum and left and right parietal regions. 75

Figure 6.10 – Accuracy from irregular x pseudo classifier. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are left and right thalamus, right precentral and lingual regions. 76

LIST OF ABBREVIATIONS

SVM. – Support Vector Machine

MRI. – Magnetic Resonance Imaging

fMRI. – functional Magnetic Resonance Imaging

rsfMRI. – resting state functional Magnetic Resonance Imaging

ACERTA. – Avaliacao de Crianças em Risco de Transtorno de Aprendizagem

BOLD. – Blood Oxygen Level Dependent

ICA. – Independent Component Analysis

KNN. – K-Nearest Neighbor

NB. – Naive Bayes

GNB. – Gaussian Naive Bayes

ROI. – Region of Interest

ReHo. – Regional Homogeneity

KCC. – Kendall's coefficient concordance

FC. – functional connectivity

rsFC. – resting state functional connectivity

MVPA. – Multivariate Pattern Analysis

IFG. – inferior frontal gyri

PA. – Phonological Awareness

TD. – Traditional Development

RD. – Reading Difficulty

AAL. – Anatomical Automatic Labeling

ANOVA. – Analysis of Variance

FEATURE SELECTION FOR NEUROIMAGING APPLIED TO WORD-CATEGORY IDENTIFICATION IN DYSLEXIC CHILDREN

ABSTRACT

Dyslexia is a developmental reading disorder characterized by persistent difficulty to learn how to read fluently despite normal cognitive abilities. It is a complex learning difficulty that is often hard to quantify. Traditional methods based on questionnaires are not only imprecise in quantifying dyslexia, they are also not very accurate in diagnosing it. Consequently, we aim to investigate the neural underpinnings of this reading disorder in children and teenagers, as part of a project that aims to unravel some of the neurological causes of dyslexia among children at pre-literacy age. In this dissertation, we develop a study of brain activation within functional MRI scans taken when children carried out pseudo-word tasks. Our study expands recently developed machine learning-based techniques that identify which type of word the study participants were reading based solely on participant's brain activation. Because such functional MRI data contains about 30,000 voxels, we try several feature selection techniques for removing voxels that are not very helpful for the machine learning algorithm. This procedure is widely used for maximizing the machine learning algorithm accuracy, and some of these feature selection approaches allowed us to achieve very accurate results.

Keywords: Functional MRI, Dyslexia, Feature Selection, Classification, MVPA.

FEATURE SELECTION FOR NEUROIMAGING APPLIED TO WORD-CATEGORY IDENTIFICATION IN DYSLEXIC CHILDREN

RESUMO

Dislexia é um transtorno de aprendizagem de leitura caracterizado pela dificuldade persistente de uma criança a aprender a ler fluentemente, mesmo apresentando outras habilidades cognitivas normais. A dislexia é uma dificuldade de aprendizado complexo e difícil de diagnosticar. Métodos de diagnóstico tradicionais, como questionários, não são somente imprecisos em quantificar a dislexia, como também não são precisos no diagnóstico. Conseqüentemente, nós visamos investigar a base neural deste transtorno de leitura em crianças e adolescentes, como parte de um projeto que tem como objetivo desvendar algumas das causas neurológicas da dislexia entre crianças em alfabetização. Nesta dissertação, desenvolvemos um estudo da ativação do cérebro com o uso de exames de imagem de ressonância magnética (IRM) funcional coletados enquanto as crianças realizavam uma tarefa de pseudo-palavras. Este estudo amplia técnicas de aprendizado de máquina recentemente desenvolvidas que identificam que tipo de palavra os participantes de um estudo estavam lendo, baseado somente em sua atividade neural. Como dados de IRM funcional contem aproximadamente 30.000 voxels, neste trabalho experimentamos com algumas técnicas de seleção de features para remover voxels que não são relevantes para o algoritmo de aprendizado de máquina. Esse procedimento é amplamente utilizado para maximizar a acurácia do algoritmo, e algumas abordagens de feature selection permitem atingir resultados muito precisos.

Palavras-Chave: IRM Funcional, Dislexia, Feature Selection, classificação, MVPA.

CONTENTS

1	INTRODUCTION	19
1.1	ACERTA	20
1.2	CONTRIBUTION	20
1.3	PUBLICATIONS	21
2	BACKGROUND	23
2.1	BRIEF OVERVIEW OF BRAIN ANATOMY	23
2.2	FMRI	24
2.2.1	PARADIGMS	26
2.2.2	FMRI DATA ANALYSIS	28
3	MACHINE LEARNING	31
3.1	FORMALIZATION	31
3.2	MACHINE LEARNING TYPES	32
3.3	USING MACHINE LEARNING METHODS	33
3.3.1	CREATING TRAINING EXAMPLES	34
3.3.2	TRAINING AND TESTING	37
3.3.3	EVALUATING CLASSIFIERS	38
3.4	MACHINE LEARNING ALGORITHMS	38
3.4.1	NEAREST NEIGHBORS	38
3.4.2	SUPPORT VECTOR MACHINE	40
3.4.3	NAIVE BAYES	42
4	FEATURE SELECTION FOR FMRI DATA	45
4.1	MOST STABLE VOXELS	45
4.2	REGION OF INTEREST	46
4.3	PARCELLATIONS	47
4.4	REHO	48
4.5	WHOLE BRAIN	49
4.6	ANOVA	50
5	RELATED WORK	51
5.1	IMPROVING CLASSIFIERS ACCURACY BY USING DIFFERENT FEATURE SELECTION METHODS	51

5.2	PREDICTING DYSLEXIA	53
5.3	MOST STABLE VOXELS FEATURE SELECTION	55
5.4	DYSLEXIA NETWORK	56
5.5	DISCUSSION	58
6	EXPERIMENTS AND RESULTS	59
6.1	DATA	60
6.2	EXAMPLE GENERATION FOR FMRI DATA	61
6.2.1	MEAN 4 SECONDS	61
6.2.2	BETAS	61
6.2.3	CONTRAST BETWEEN CONDITIONS	62
6.3	SINGLE SUBJECT EXPERIMENTS	63
6.4	CROSS SUBJECT EXPERIMENTS	64
6.5	DISCUSSION	65
7	CONCLUSION	77
	REFERENCES	79
	APPENDIX A – Classification accuracy in each ROI	85

1. INTRODUCTION

Dyslexia is a neurobiological disorder that affects one's reading and writing skills. Basically, people with dyslexia have difficulty mapping the glyphs of a word with its sound. Consequently, they have difficulty in decoding words (pronouncing printed words) and encoding words (spelling words). Because dyslexia is not a comprehension or intelligence problem, but rather a problem in reading and spelling words, people with dyslexia do not have trouble understanding texts when they are read these texts out loud, always having listening comprehension skills higher than reading and writing skills [Sha08]. The diagnosis of dyslexia involves a complex, multidisciplinary evaluation of reading performance; cognitive abilities and intelligence; and school and medical history. It takes at least two years of regular schooling before a child can be diagnosed with dyslexia, and most children are diagnosed from 8 to 9 years, (see DSM-5 criteria ¹), Thus, identifying early indicators of children at risk for learning disabilities, such as dyslexia, may help understand early signs of reading impairment and help children develop strategies to cope with this condition.

However, studies have shown that no standard test used today is able to detect or predict dyslexia precisely [HMB⁺11]. Thus, new tests are needed in order to fill this gap.

One of the most important new techniques used for identify neuropsychological disorders is Functional Magnetic Resonance Imaging (fMRI), which is an neuroimaging method that indirectly measures neural activity over time. Given its non-intrusiveness to patients, fMRI is widely recognized as a powerful diagnostic tool for conditions with a neurological basis. Many other neuropsychological disorders such as autism [JKM⁺12], Alzheimer's disease [WSN⁺09] and dementia [WS12] have been successfully identified using Functional Magnetic Resonance Imaging (fMRI), Here, some neural activity patterns captured by fMRI data are known to indicate a person's cognitive state or the presence of a neuropsychological disorder. Recent research shows that fMRI data, in combination with machine learning techniques, can be used to predict the cognitive state of subjects [HR06] [MSC⁺08] [SMM⁺08]. Learning disabilities such as dyslexia may be investigated using fMRI to identify the differences in brain function that may underlie such developmental difficulty. To our knowlege, there is no work on applyng fMRI data to identify dyslexia.

This work is organized as follows: Chapter 2 describes the application domain in which we want to use classification, that is fMRI and brains; Chapter 3 describes the applicable machine learning algorithms used in our research; Chapter 4 describes feature selection methods we use in the fMRI data; Chapter 5 describes and compares related work on fMRI and machine learning; Chapter 6 presents the experiments and results; finally, we draw conclusions and indicate potential future work in Chapter 7.

¹The Diagnostic and Statistical Manual of Mental Disorders has guidelines for diagnosing someone with a mental disorder, such as dyslexia [A⁺13]

1.1 ACERTA

This study is part of the ACERTA project, which stands for Evaluation of Children at Risk for Reading Difficulties. The goal of the project is to understand the differences that underpin the inability of dyslexic children to learn to read fluently, in comparison with their normal reading peers. To achieve this goal, the project applies Functional Magnetic Resonance Imaging (fMRI) to obtain brain-imaging data from children with dyslexia and controls. In this work, we take the first steps towards our overarching objective, which is to use brain imaging data from children with dyslexia to identify specific cognitive states associated with reading impairment. For this purpose, while in the MRI scanner, children perform a task in which they read 3 types of words. Regular words, which we write as we say them (e.g. *dinheiro*, *gelatina livro*), irregular words, which we write them differently than we say them (e.g. *sorte*, *taxi*, *exemplo*) and pseudo words, which are words that appear to be in Portuguese but have no meaning (e.g. *laberinja*, *prina*, *cusbe*).

1.2 Contribution

There are many difficulties in using classification with raw fMRI data, as the amount of data generated by a single scan is massive. Consequently, we need to find ways of reducing the amount of data using feature selection techniques and reaching a reliable classification. The main contribution of our work is to test and report on different feature selection techniques to improve the classification accuracy when analyzing fMRI data of dyslexic children. We empirically test these techniques using fMRI data from children from the ACERTA project and show that we can use classifiers for discover which type of word they are reading as well as the most important brain patterns for the classifier to discriminate the category of words. Therefore, the contributions of this work are:

- We used classification and feature selection methods with fMRI data of children with dyslexia performing a reading task. Reading brain networks are distributed all over the brain, specially in children with dyslexia. Thus, show how we identified distributed brain patterns using fMRI data.
- We find a way to process noisy data and generate cleaner examples for classification. For this purpose, we transform fMRI data into contrast images.
- Test four feature selection methods and discuss how each one deal with distributed brain patterns.
- Generate a classifier that can generalize among study participants and yields very good accuracies. The accuracy gets better depending on the feature selection we used.

- We discovered the most important brain regions for the classifier to identify which type of word children are reading. The brain regions belongs to reading neural networks from both traditional and dyslexic readers.

1.3 Publications

During this work, we published the following papers:

- FROEHLICH, Caroline; AURICH, Nathassia; MENEGUZZI, Felipe; BUCHWEITZ, Augusto and FRANCO, Alexandre R. Categorical and dimensional variable prediction from state fMRI data, a new example and tutorial for NiLearn. Brainhack Unconference and Hackathon, Sèvres, France, 2013
- FROEHLICH, Caroline; MENEGUZZI, Felipe; FRANCO, Alexandre R. and BUCHWEITZ, Augusto. Classifying Brain States for Cognitive Tasks: a Functional MRI Study in Children with Reading Impairments, In Proceedings of the 24th Brazilian Congress on Biomedical Engineering (CBEB), Uberlândia, MG, Brazil, 2014.
- FROEHLICH, Caroline; MENEGUZZI, Felipe; FRANCO, Alexandre R. DRESCH, Luiz and BUCHWEITZ, Augusto. Identifying the neural representation of word reading in children diagnosed with dyslexia, In Organization for Human Brain Mapping, Honolulu, Hawaii, 2015.

2. BACKGROUND

As this is a multidisciplinary work that includes brain functions, neuroimaging, and machine learning, we review important brain parts in Section 2.1 as well as key points of fMRI imaging data in Section 2.2.

2.1 Brief overview of brain anatomy

We start by describing the physical areas of the brain, later focusing on areas of particular interest to this work, namely, language related areas. We start with a major brain division, which are the hemispheres. The brain is divided in the left and right hemisphere. Both hemispheres have the same architecture (both contain the same brain divisions), but each brain region in each hemisphere have distinct functions. Some examples of brain location in left and right hemispheres performing different tasks are: first, the motor area in the left hemisphere controls the right part of the body, while the motor area in the right hemisphere controls the left part of the body; second, the language area, which is typically located in the left hemisphere in most of the right handed people, while the same areas in the right hemisphere have different functions [Deh09].

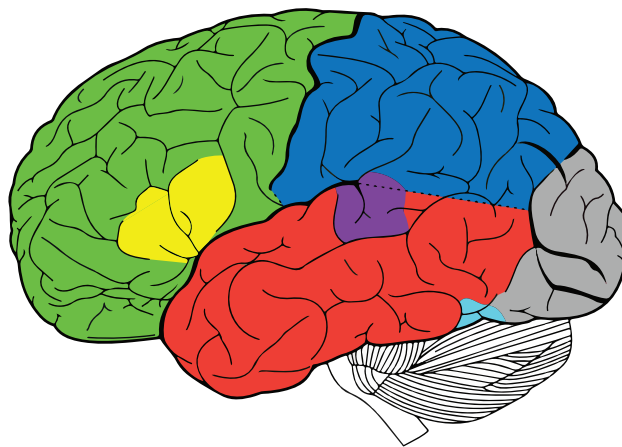


Figure 2.1 – The 4 lobes of the brain. The configuration of the lobes in both hemispheres are replicated. The frontal lobe (green) is related with planning functions. The temporal lobe (red) is related to language functions and giving meaning to visual inputs. The parietal lobe (blue) is related to processing sensory inputs and contains the motor areas. The occipital lobe (gray) is related to visual processing. The two important language-related areas are the Broca's area (or inferior frontal gyrus) in yellow, related to speech production and the Wernicke's area (or superior temporal gyrus) in purple related to speech processing.

The brain is further divided into four lobes, each one of which is specialized in a specific task. This subdivision is illustrated in Figure 2.1¹, which shows a left view of a brain with its four lobes. Some important language regions in the brain are the Broca's area (yellow) and the

¹This image was kindly provided by Anibal Solon

Wernicke's area (purple). Broca's area is located in the bottom of the inferior frontal part of the brain (inferior frontal gyrus), and is related to speech production. People with a lesion in this area are able to understand what other people write and speak, but sometimes are not able to produce spoken and written language. Wernicke's area is located in the back of the superior temporal part of the brain (superior temporal gyrus), and is related to speech processing. People with lesion in this area are able to write and speak, but do not understand what other people write and speak.

However, these are not the only areas involved in language processing and production. For example, the occipital lobe, which process visual input, also processes written symbols and decode them. Additionally, people with reading difficulties need more processing for properly reading. Thus, they request more brain areas than traditional readers. Instead of using just Broca's area in the left hemisphere, they recruit the same brain region in the right hemisphere as well as some other regions in the frontal lobe.

2.2 fMRI

Neuroimaging involves different techniques to acquire images of the brain, each of which have distinct purposes, such as measuring a subject's brain activation patterns or showing views of a subject's anatomical brain structure. Each neuroimaging exam has a distinct spatial and temporal resolution, and detect different tissues, and highlights different physical processes (e.g. physical, chemical, structural) taking place in the brain. For example, structural MRI (Magnetic Resonance Imaging) is a neuroimaging scan that shows a static view of the brain anatomy in detail. MRI data has a good spatial resolution, showing with millimeter accuracy brain tissue of white matter and gray matter. With the MRI scanner we can acquire a high resolution 3D image of the brain showing these tissues.

For this project we are interested in fMRI (functional Magnetic Resonance Imaging) [HSM04], a 4D functional neuroimage that consists of a time series of 3D images of the brain. While MRI acquires one high resolution image showing the brain structure, fMRI acquires many low resolution 3d images in order to detect the activation of brain regions over time, since its purpose is to map brain activity [BZYHH95]. Compared to EEG (electroencephalography), another technique that can measure brain activity, fMRI data has a high spatial resolution (it shows the brain details at millimeter scale, similar to MRI), and a low temporal resolution (because it takes the MRI scanner up to a few seconds to acquire each image).

While the purpose of fMRI is to detect neural activity during experiments, it does not measure neural activity directly. Instead, it measures changes in the magnetic properties of the blood when neural activity occurs. More specifically, it measures the magnetic properties of hemoglobin, which is a blood component that carries oxygen from respiratory organs to the other body parts. While deoxyhemoglobin (hemoglobin without oxygen) creates negative local magnetic resonance signal, oxyhemoglobin (hemoglobin with oxygen), does not alter the local signal. When someone is

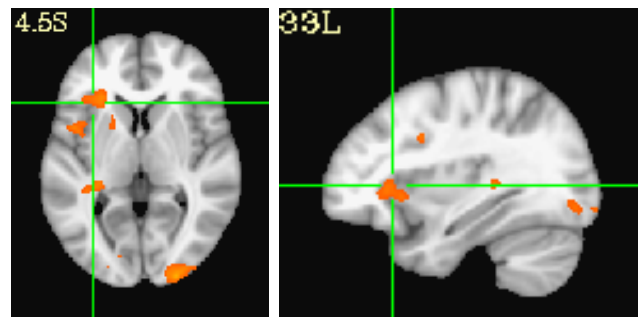


Figure 2.2 – Differences between brain activation when children are resting and performing the pseudo-word task. The orange areas indicate where there is ' activation in the pseudo-word task than in rest across subjects.

performing a task, the local magnetic signal in the activated area increases. That is because that area needs more oxygen, then the oxyhemoglobin level increases in order to to supply oxygen. Figure 2.3 illustrates how the oxihemoglobin level changes when some brain area has neural activity. Since the region gets more oxyhemoglobin, that region has more oxy than deoxyhemoglobin, increasing the local magnetic resonance signal (deoxyhemoglobin lowers the local signal. So, if we have less deoxyhemoglobin, the local signal increases) [Maz09]. We call this measure of the level of oxy and deoxyhemoglobin BOLD (Blood Oxygen Level Dependent). Figure 2.4 describes the changes in the BOLD signal over time.

In MRI, we can observe the brain region's anatomy in detail. In functional MRI, we can observe which areas of the brain are functioning together, i.e. which brain region is activated when performing a specific task and how brain regions work together. For example, we can test which parts of the brain increase their activity when a subject is performing a task while inside the MRI scanner, such as moving the left hand or checking a word is real or not. Figure 2.2 is an example of a task-fMRI experiment that shows brain areas activated when children are reading words. These areas are activated compared to when the children are resting inside the scanner. fMRI scans allow us to infer multiple information from the workings of the brain, two examples of what we can infer from the study of our example are: First, the brain areas related to the cognitive task. Second, it indicates how these brain parts interact: if they work synchronously (both regions are activated at the same time), when one is activated, the other deactivate, or if they are not related (the activation of one is not dependent on the other).

We can perform many analyses with the information provided by fMRI data. First, we can see the brain function when someone is performing a cognitive or motor task, mapping the regions where there is more neural activity. Second, we can identify cognitive disorders using fMRI data. We can find brain activation patterns in patients with the same cognitive disorder by looking for brain regions that are activating or communicating differently between healthy people and patients. These patterns form biomarkers, that in this context are a set of characteristics (brain region activations) that identify a cognitive disorder.

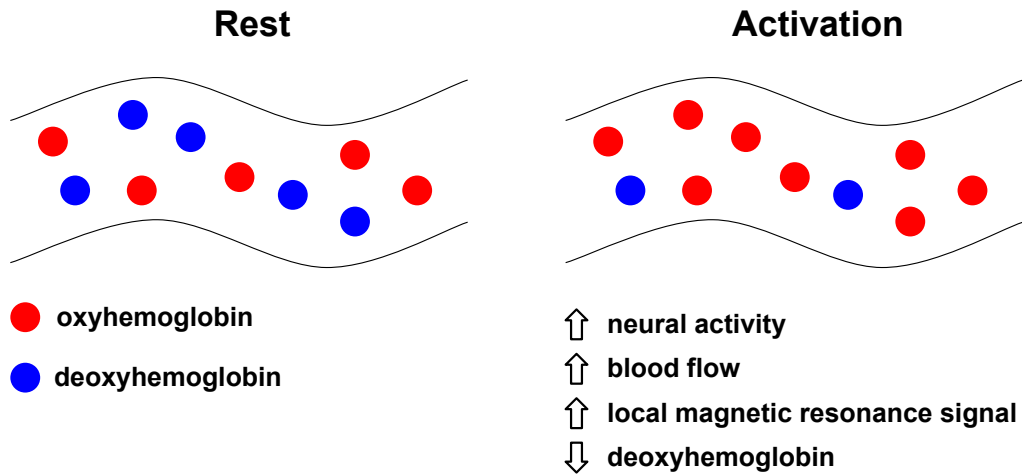


Figure 2.3 – Differences in the blood flow when a brain region is in rest and activated. In rest condition, we have the same level of oxy and deoxyhemoglobin. Conversely, when there is neural activity, the blood flow increases along with the oxyhemoglobin level. That happens in order to supply local oxygen demands. As the oxyhemoglobin level increases, the deoxyhemoglobin level decreases, which makes the local magnetic resonance signal increases too. Inspired in [Maz09]

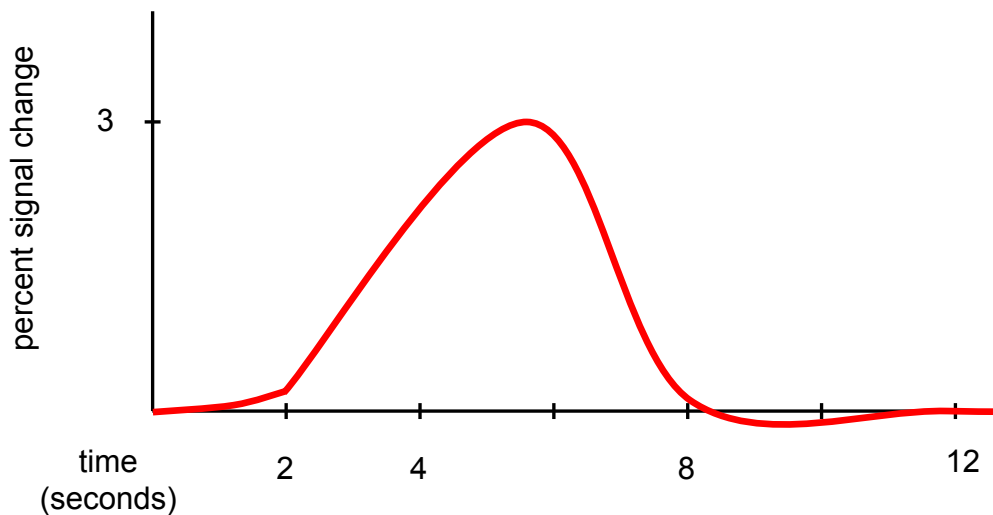


Figure 2.4 – After brain activity, there is immediate oxygen consumption; 2 seconds later, more oxyhemoglobin arrives in the activated area in order to supply the oxygen demand. As there is more oxygen (and oxyhemoglobin) than the required area needs, the local magnetic resonance signal increases. The signal reaches the maximum value at 5 seconds, and subsequently returns to the baseline value at 12 seconds. The BOLD signal changes only 3% when a brain area is activated.

2.2.1 Paradigms

fMRI can answer questions about psychological processes, such as the questions from the ACERTA project which asked which areas of the brain children with dyslexia use to read. For this purpose, we need a well defined set of tasks for the children to perform inside de scanner that can discriminate this network. This set of tasks is called paradigm or experimental design [HSM04]. The

main goal of the paradigm is to use a stimulus that engages subjects in a specific mental process in order to measure the changes in the BOLD signal caused by the stimulus in an organized and easy to analyze way.

A stimulus can be of various types, such as to recognize whether a picture on the screen shows a face or an object, whether a word is real or not, if a spoken word is a noun or a verb or if the lights are on or off. The stimulus is expected to activate a brain region necessary to process it, consequently changing the blood flow and rising the BOLD signal. In this way, we can discover which parts of the brain the stimulus activates and how strong the BOLD signal is.

We can create a set of stimuli of a specific type (a.k.a. condition). For example, in a motor study, the conditions can be moving the right hand and/or moving the left hand. In another study about the difference between the neural representation of nouns or verbs, the conditions are showing nouns and showing verbs.

We expect that the same stimuli (i.e. moving the right hand) always shows a greater neural activity in the same regions, while different stimuli (i.e. moving the left hand) shows neural activation in another region. Thus, stimuli of the same conditions show a greater neural activation similarity while stimulus of different conditions are supposed to show neural activation differences. The neural activation differences between conditions are the way we discover brain areas involved in a specific task.

We divide the paradigm conditions in two types, task condition and baseline condition. The task condition is the most relevant condition for the study, which is the task we really want subjects to perform. Conversely, the baseline condition establishes a baseline to compare with the task condition. Because the brain never ceases activity, data from the baseline condition shows voxels that are activated when subjects are not performing any task in particular (or at least a task that is not being prompted by the selected stimulus). Therefore, we subtract baseline condition data from task data to emphasize the voxels involved in the task.

Besides comprising various conditions, a paradigm also specifies the number of times each condition is shown to the subject during the experiment. Each time a condition is shown to the subject is called a trial. For example, in a study about verbs and nouns, we can present as stimulus a list of different verbs and nouns. The paradigm can be divided into blocks, where each block groups various trials. For example, we can have a block with 10 nouns and another block with 10 verbs in a paradigm. Usually, we have a baseline condition block between two task blocks in order to clear subjects mind and minimize the interferences between two contiguous blocks. The duration of each block can vary from several seconds to a few minutes.

There are two types of task paradigm depending on how the blocks are organized. The first one is block design, which shows several trials of the same condition in one block. Figure 2.5 shows a piece of a block design paradigm where the same condition (word) is shown several times in a block, followed by a brief rest period. We expect the subject to have the same neural response in every trial of the same condition, so we can model how the brain response to that condition is. The second one is event related, which mix different task conditions in the same block. Figure 2.5

shows a piece of a even related block where the different conditions are shown randomly in the same block.

An additional paradigm which uses no task is being increasingly applied in fMRI studies, the resting state fMRI (rsfMRI). rsfMRI experiment has no task, instead the subject is required to look at a black screen for about 7 minutes, and to do not think in anything in particular. Although rsfMRI does not seem immediately useful as the task-based experiment, because we do not know what subjects are doing or if they are thinking in something in particular, however, this experiment can tell us how the functional connectivity of the brain is, that means which brain regions are communicating to each other brain regions and how they are communicating. This information is used to detect biomarkers that characterize cognitive disorders [CHHM09].

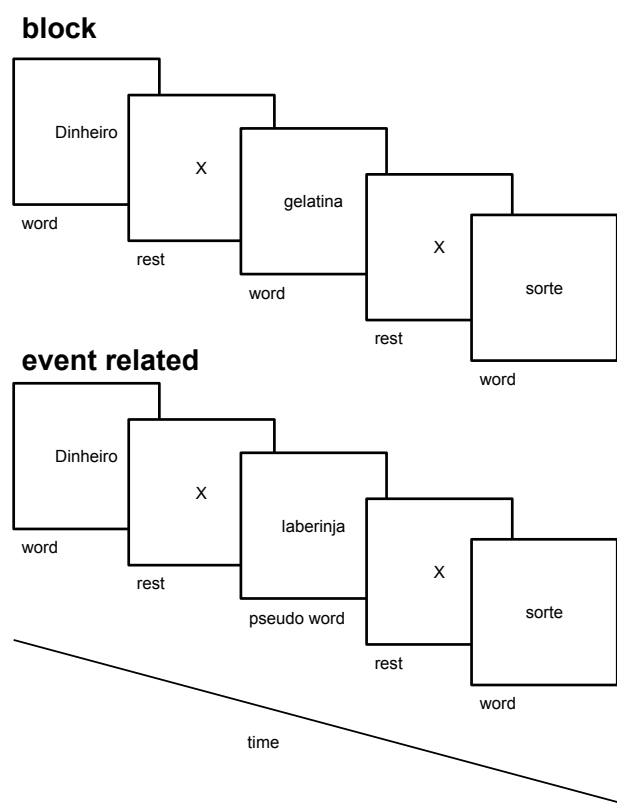


Figure 2.5 – Schematic example of 2 different paradigm blocks using the pseudo-word task. In the block design paradigm, just one condition is presented in a single block, while in event related all conditions are presented randomly in a single block. All trials are followed by a few seconds of rest.

2.2.2 fMRI data analysis

After creating the paradigm, running the imaging sessions and preprocessing data, there are a number of methods we can use to infer useful information from such data. We can perform various statistical analyses to extract activation patterns in the data of a single patient or a group of patients (using statistical analysis is more common than using machine learning to analyse fMRI).

Statistical methods select voxels that show a statistical difference between distinct groups. These voxels are used to generate images showing which region of the brain has similar activation. It generally uses a statistical significance test to score each voxel separately and selects the voxels with higher scores.

In this context, a group can refer to a set of trials of the same condition a single patient performed or a group of subjects with the same cognitive disorder. In summary, a group contains similar observations that are intended to show different neural activation from the other groups. Thus, we can analyse data from single patients as well as data obtained from a group of patients performing the same paradigm. Figure 2.2 shows an example of a statistical analysis discriminating the activation between two different conditions in the pseudo-word task: when patients are viewing a pseudo-word (task condition) and when they are resting (baseline condition).

The statistical methods for analysing fMRI data, also called univariate methods, generally work as follow. Univariate methods take the same voxel or brain part of the entire group 1 and look for differences between these voxels and the corresponding voxels or brain parts of the entire group 2. Therefore, univariate methods look for differences between groups analyzing one part of the data of the hole group at time. Finally, they calculate the differences between each feature of each group using the t-test method.

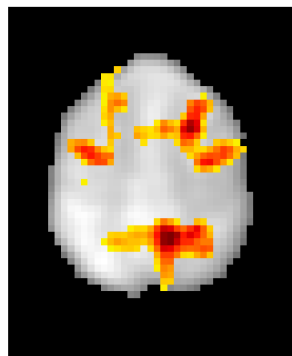


Figure 2.6 – ICA algorithm applied in rsfMRI data, generating 20 components. The image is generated with the scikit-learn [PVG⁺11] implementation of ICA

There are many univariate methods, a widely used method is Independent Component analysis. Independent Component Analysis (ICA) [CAH⁺03] methods separate different sources or components of signals that are mixed in the same signal. ICA assumes that each signal source is independent and its distribution is non Gaussian, and uses statistical methods to separate the signals. Figure 2.6 shows a result of the ICA algorithm applied to rsfMRI data. The highlighted areas are clusters of voxels with statistical similarities.

3. MACHINE LEARNING

Machine learning is a branch of artificial intelligence that aims to make programs that improve performance from data. It includes algorithms that improve their own performance at some task through experience, getting better on the task as they receive more and more examples to learn from. Machine learning algorithms can be used to perform many tasks humans can, except that they process much more data in a fraction of the time because machine learning algorithms are better in data intensive tasks. For instance, they can learn to differentiate between spam and ham emails, and classify emails as being spam or not [Alp04].

There are three types of machine learning, the main difference between them is the type of data they take as input, each of which changes the process they require to extract information about the examples. The first is unsupervised learning: in this approach, the algorithm learns using only the examples that are provided. Because we do not have much control over the learning process, as we just provide data, these methods make it more difficult to predict what is being learned. The second is supervised learning: in this approach we provide examples to the algorithm with labels of what the given example is, so we can control more easily what we want it to learn. The third is reinforcement learning: we use this kind of algorithm in a specific scenario, where an agent is required to choose actions and refine its own choices over time based on how its past choices were evaluated.[TSK05]

In a context where we have a software that deals with lots of data, machine learning algorithms can be the intelligent part of it, discovering things about these stored data, or improving modules that are difficult to program. Some modules are hard to program because they have many rules and exceptions in a way that humans find hard to understand and build. In contrast, an approach that learns all this knowledge may provide better results and be easier to develop.

3.1 Formalization

We now formalize machine learning before describing specific methods in Section 3.4. In the training process, the algorithm tries to formalize the concept given in the examples (we call the set of examples *training data*), learning a function that defines it (much like function fitting). The complexity of inputs and outputs induces a hypothesis space, that consists of all the possible functions that map the inputs to the outputs. Russel [RN10] shows that we can apply these algorithms in simple applications: given a training set of points generated by an unknown function f , find a function h that is an approximation of f . h is a hypothesis that tries to generalize how the points are generated.

Machine learning is formally defined by Mitchel [Mit97] as: “an algorithm that learns from a set of experiences E with respect to some class or task T and performance measure P , if its performance at T , as measured by P , improves with experience E .”. In other words, T is the task

the algorithm is required to perform, for instance, classify an email as spam or ham, or in the function example, h is the function that performs T . E are the examples the algorithm receives in order to learn the task; in the function fitting example, the examples are the points generated by f , and in the task of classifying emails, we can provide emails already classified as spam or ham as examples. P is how we measure the ability of the algorithm in the learned task, it tells the algorithm how it performed and allows it to modify and improve its behavior according to this metric to future tasks.

There are many problems we can solve with machine learning, such as working with continuous variables using regression, predicting the continuous value of an instance, or using clustering for grouping the examples in the training set by similarity. However, we focus in the supervised learning classification problem, where we have a discrete and finite set of categories, and assign some category to new instances.

As an illustrative example adapted from [Mit97], we consider the problem of classifying emails as spam or ham in more detail, where L is the learning algorithm that creates a classifier for the emails. D_C is the training data of class C . We say that T is the email classifier, D_C are the experiences E and P is the accuracy of the classifier. D_C is a collection of emails already classified, in which the class C represents the possible values the algorithm can give to an email, $\{Spam, Ham\}$. In this formalization, the training data is expressed in the form $D_C = \langle x, c(x) \rangle$, where x is an instance (email), and $c(x)$ is the classification for the instance, $c \in C$. We use D_C to train L to distinguish *Spam* and *Ham* emails. L , in turn, is used to create an email classifier after the training phase. Therefore, L creates a hypothesis h about C , that can classify new instances x_i returning a class c , that is the class the algorithms says x_i belongs.

As a consequence, the output that L generates is inductively inferred from D_C and x_i , resulting in an inference about x_i that will probably not be correct in general. This occurs because the algorithm requires a set of assumptions to deduce the output based on its inputs, even if the new instance x_i was never seen before (there is no instance like x_i in the training data). Consequently, the deductions of the algorithm are dependent of the training data, and it has a probability of giving the right answer (we can measure the accuracy of the algorithm).

3.2 Machine learning types

Since the input given to machine learning algorithms defines how they work, we need to formally model the three types of learning algorithms we mentioned before. Although the generic learner L always generates the same output c , that is a classification for the new instance x_i , each type of approach processes differently the training data D_C , because they have to adapt to the different structure of D_C [Mit97].

In unsupervised learning, the algorithm learns how to classify data by itself, when we provide as input only a large training data set. The inputs contain unlabeled data, and it does not know what classes are in the training data, it is in the form $D_C = \{x\}$. In this approach, there are

no predefined classes $c \in C$, so the algorithm creates C and splits the training data according to these new categories, looking at relevant differences between the instances of D_C [RN10].

An example of this type of machine learning is a neural network algorithm that is used to discover high-level concepts in a training data containing images taken from Youtube videos [LRM⁺11]. It successfully recognized cats and human bodies, even though it was not told these two classes were in the dataset.

In supervised learning, in order to train the algorithm, we show each instance of the training data and the class it belongs to. This approach is divided into two phases. First, we train the algorithm to recognize each class of C , providing to it the training data. An instance in D_C is a tuple $\langle x, c(x) \rangle$, where x is the input and $c(x)$ is the classification for x . Finally, we ask the algorithm to classify new unlabeled data, x_i . If we provide to L a big enough training dataset, with few misclassified data, and with different and useful examples, the probability of L to give the right answer for x_i increases. Consequently, for any learning algorithm, its accuracy depends on the training data, so we have to choose carefully the examples in D_C [RN10].

The email classifier described in the previous section is an example of supervised learning, as the algorithm learns what is *spam* by the emails examples classified as *spam* or *ham*. After processing the training examples, we can ask it the classification of a new email.

In reinforcement learning, the agent performs an action and receives feedback that measures the outcome of the action, receiving a positive feedback (reward) if it is a good and a negative feedback (punishment) otherwise. Therefore, the learner refines the choices it makes by collecting the feedback of past actions, returning a policy with the best moves it has found so far [RN10].

This approach has many applications in games. If we let the algorithm play, it can learn the game rules and the best moves to choose in each scenario. The algorithm also can adapt its policy quickly when the game changes [Gho04] [GHG04], [Tay11].

3.3 Using Machine Learning Methods

Using machine learning algorithms is not just about building one classifier. There are many data transformations and measures we have to perform in order to create a usable classifier. [PMB09] We describe the 3 steps we follow in the experiments Chapter in order to train and measure how good our classifier is. First, we generate usable examples transforming data in feature vectors (transforming data into feature vectors, using feature extraction and preprocessing them). Second, we train the classifier with examples and test its accuracy. And third, we measure how well the classifier performed. A complete description of each step is listed below.

3.3.1 Creating training examples

Transforming data into a feature vector

Data used for machine learning can be of various types, such as images, texts or data from tables. We want to work with a representation of the data that can be processed by these algorithms, called feature vector. In the task of classifying emails, we can represent each instance using a dictionary that contains relevant words, where each word is a feature that counts the frequency of the word in a given email. We choose words that are relevant to the learning task at hand. Thus, we can represent an email using a dictionary counting the frequency of some given words. For instance, the dictionary d has two words, *test* and *email*:

$$d = [\textit{test}, \textit{email}]$$

If we have two emails, $e1$ and $e2$:

$e1$ = This is a test. We are testing the email server. The email server will be unavailable until tomorrow.

$e2$ = Test the best email server for free for two years.

The correspondent feature vectors of each email, $fv1$ and $fv2$ is:

$$fv1 = [2, 2]$$

$$fv2 = [1, 1]$$

We can see a feature vector as a point in a Cartesian plane, where each feature is an axis or a dimension, and all the dimensions in the feature vector form a state space. We can draw $fv1$ and $fv2$ in a Cartesian plane, where x axis is the frequency of the first word in d , *test*, and y axis is the frequency of the second word in d , *server*, as we show in Figure 3.1.

If a feature vector is a point in a Cartesian plan, we can calculate the distance between two points. The distance between $fv1$ and $fv2$ may be calculated using the manhattan distance. Other possibilities to calculate distances are described in Section 3.4.1.

$$(2 - 1) + (2 - 1)$$

$$= 2$$

We can try to improve the classifier's accuracy by adding more words to the feature vector, adding two words to d , *server* and *trial*, obtaining the dictionary d' :

$$d' = [\textit{test}, \textit{mail}, \textit{server}, \textit{trial}]$$

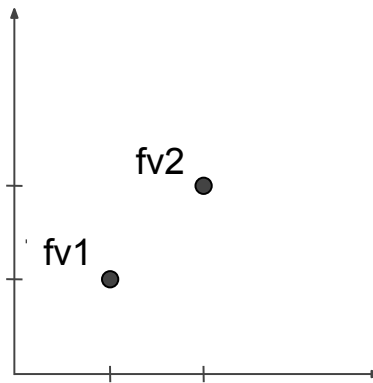


Figure 3.1 – $fv1$ and $fv2$ graphic, where x axis is the frequency of the word *test* and the y axis is the frequency of the word *server*

$fv1$ and $fv2$ are updated according to d' to $fv1'$ and $fv2'$:

$$fv1' = [2, 2, 2, 0]$$

$$fv2' = [1, 1, 0, 0]$$

Calculating the new distance between $fv1'$ and $fv2'$ we have:

$$\begin{aligned} (2 - 1) + (2 - 1) + (2 - 0) + (0 - 0) \\ = 4 \end{aligned}$$

So far, three things happened when we added more features. First, the state space gets larger because we increased the number of dimensions. Second, the distance between the two points $fv1$ and $fv2$ increases. Third, we added a useless feature *trial*. *trial* does not change the distance between $fv1$ and $fv2$ because it does not appear in any email, but it increased the size of the state space, as *trial* becomes a dimension in the cartesian plane. Consequently, adding more features to the feature vector is not always good, because as we enlarge the state space, the distance between points increases, and we increase the chance of having useless features.

This issue is called curse of dimensionality and occurs in high dimensional state spaces: the points in space become more distant as we add more dimensions, because the state space grows quickly, making all points very distant one from another. That means we need many more emails to cover all of the state space, because we need many more examples to cover a large enough portion of the state space. Domingos [Dom12] shows that covering all the state space is unfeasible due to the required amount of data. If we have a state space with 100 dimensions and a trillion examples, these examples cover only 10^{-18} of the state space. Therefore, huge feature vectors are not always good, because they can cause the curse of dimensionality and add irrelevant features. Increasing the size of the feature vector can also add noisy features, if we have common words that will always appear in any email, or redundant features, if we have different words with similar meaning in the dictionary.

Feature selection

We have seen that even after transforming data in examples by using the feature vector, these examples may contain uninformative features or become too large. We address these problems by reducing the number of features using feature selection techniques [HK00]. Considering that all examples are in a matrix, where each row is an example and each column is a feature. Feature selection methods take as input this matrix and output a matrix with the same number of examples but with fewer columns.

One way of doing that is by manually selecting features that are relevant to the task at hand. In the email classifier example, in the case where there are several words in the dictionary, we could get rid of words of restricted domains that usually do not appear in emails, or ignoring common words that will probably appear in every email, such as “for” and “to”.

In other domains where we cannot decide explicitly how important a feature is, scoring/filtering and wrapper methods are used [GE03]. The first method gives a score for each feature using a given criterion, evaluating each feature independently. Then, the algorithm selects only the features that have the best scores. Typically, filter methods use an univariate criterion (t-test) to evaluate features, and are independent of the final classifiers accuracy. The second method recursively selects a subset of features until find the best subset. It evaluates the accuracy of the classifier using the selected features and tries to improve the classifier accuracy by changing this subset. This method is time consuming because it has to evaluate all possible subsets of features, making it impractical to use without heuristics.

Preprocessing examples

The last step we can perform in order to acquire better examples is to preprocess them. We usually do it when features have a continuous value. This process reduces the differences between the features, minimizing the problem of one feature having much more importance than the others. One way of doing that is to normalize all matrix feature vectors by normalizing each row separately, so each row has mean 0 and a standard deviation equal to 1 [PMB09]. In this context, the normalization of a feature vector means subtracting the feature vector mean of each feature and dividing it by the standard deviation of the feature vector.

For example, if we want to normalize the feature vector $fv = [5, 8, 34, 6, 9, 14, 88, 80, 19, 7]$, we calculate that the mean of fv is 27 and the standard deviation is 31, and subtract the mean of each feature and divide it by the standard deviation, obtaining the normalized feature vector $fv' = [-0.7, -0.7, 0.2, -0.6, -0.5, -0.4, 1.9, 1.6, -0.2, -0.6]$.

3.3.2 Training and testing

When the algorithm and the data have been preprocessed, we have to choose a set of instances in the dataset for the training phase, that is the training set. After the training phase, we test the algorithm using instances of the dataset again in order to measure the accuracy of the learner, that is the percentage of correctly classified instances in a given test set. If we use the same data to train the algorithm and to test it, the success rate should be 100%, because it has previously seen the data. Therefore, we show to the algorithm data it has never seen before to evaluate if it learned successfully. There are several methods to split the training data in the training and testing sets, preventing us to test the accuracy of the algorithm with the data the classifier has used to learn [HK00]. We describe two methods to split data: cross-validation and bootstrapping. Cross-validation is widely used to test classifiers accuracy while bootstrapping is used as part of other algorithms, so we describe cross-validation in more detail than bootstrapping.

The first method is the most common approach for training and testing with different instances [HTF01]. In the cross validation method, the dataset is equally divided in k folders, and we run the method k times, keeping one different fold as the test set for each run, and using the other folds as the training set. The algorithm's accuracy is the average accuracy for each run, and the number of folds are usually 10 or 5 (using 10% or 20% of the dataset for testing). For example, dividing D_C in 3 folds, D_{C0} , D_{C1} and D_{C2} , in the first run we keep D_{C0} as the test set and train the machine learning algorithm with D_{C1} and D_{C2} . In the second run we keep D_{C1} as the test set and the training data is D_{C0} and D_{C2} , and so on, until we run the algorithm 3 times and calculate the average accuracy. There is a special case of cross-validation called *leave-one-out*, where k is equal to the number of instances. Leave-one-out uses all the data to evaluate the classifier, so it is evaluated more times than others values of k . The accuracy of the generated classifier depends on the training data size, having more examples is better because it eliminates noise and variability. The accuracy also depends on how the examples of each class are distributed in the training data. It is better to have the same number of examples of each class. If we don't, the classifier tends to predict new examples of being of the class with most examples. Similarly, when using cross-validation, each fold must contain examples of all classes [PMB09].

The second method is bootstrapping [Joh01], it selects the test set sampling the dataset with replacement. That means examples chosen once to be part of the training set are likely to be selected again. The method works as follow: the dataset has n examples, bootstrapping selects n times an example to be part of the training set, and one example can be selected many times. The selected examples are the training set and the not selected ones are the test set. There are many bootstrapping methods, the most used is the .632 bootstrap method, that selects 63,8% of the examples to be part of the training set and 36,8% of the examples to be part of the test set.

3.3.3 Evaluating classifiers

In order to evaluate if a classifier is doing well in the testing set we measure its accuracy, that is how much times the classifier correctly identified the class of new examples. But we have to ensure that the classifier performs well in examples outside the dataset. Therefore, we measure its true accuracy, that is the probability the classifier chooses the right class for a new example, if this example was generated using the same distribution of the test set [HTF01].

We also can ensure that a classifier c is statistically significant and is not randomly guessing the classes of new instances. We demonstrate it by proving that an hypothesis such that c *correctly classifies new instances 95% of the time* is correct. In statistics, we work with the null hypothesis, that is the denial of the original hypothesis: c *does not correctly classifies new instances 95% of the time*. We accept the original hypothesis by refusing the null hypothesis, using the t-test for proving it. t-test statistically accepts or rejects the null hypothesis, proving that the classifier is correct for most of the data. This is based on the assumption that the data have a normal distribution and form a bell curve, therefore, c is correct for most of the data, but it may be wrong when data belong to the edges of the bell curve. We set how much we accept the classifier is wrong by setting the p-value. In the hypothesis, c is wrong 5% of the time, so the p-value is 0.05.

3.4 Machine Learning Algorithms

After describing abstract machine learning algorithms, we present instances of these algorithms.

3.4.1 Nearest neighbors

K-Nearest Neighbor (KNN) is a simple machine learning algorithm that predicts the class of a new instance looking for the closest instances in the dataset. It returns the k most similar examples in the training data to a new instance. More formally, it finds the k examples in the training data D_C nearest to the new instance x_i . We find the k nearest neighbors of the instance x_i using the notation $NN(k, x_i)$.

KNN represents the input space as a Cartesian plane, where an instance is a point in space represented by a feature vector. The number of dimensions of the state space is the same as the number of features there are in the feature vector. In order to find the nearest neighbors of the new instance x_i in the Cartesian plane, we measure the distance between x_i and all other examples in D_C . There are multiple methods to calculate the distance between x_i and an arbitrary instance x_j , $x_j \in D_C$. One possible way of calculating this distance is using the Minowski distance, that calculates how far x_i is from x_j considering all dimensions of the instances. We write a feature

vector as $[f_1(x), f_2(x), \dots, f_n(x)]$, where $f_n(x)$ is the n th feature of instance x , and the distance between x_i and x_j is

$$L(x_j, x_i) = (\sum |f(x_j) - f(x_i)|^p)^{1/p}$$

The formula basically calculates the differences between each feature of the two instances and sums all these differences. It is possible to adjust the formula to calculate the distance between x_i and x_j in different ways, obtaining well known formulas to measure distances by setting the variable p . By setting $p = 2$, we obtain the Euclidean distance. Setting $p = 1$ we obtain the Manhattan distance, that is the sum of absolute differences between Cartesian points. And the Hamming distance by counting the number of different features, that is used to measure the distance between boolean vectors

In Figure 3.2 we want to classify the new instance x_i in gray by measuring the distance between x_i and all other instances. We start by calculating the distance between x_i and x_j . Using the Manhattan distance, we have:

$$\begin{aligned} L(x_j, x_i) &= |1 - 4| + |1 - 5| \\ &= 7 \end{aligned}$$

And using the Euclidean distance, we find a different value:

$$\begin{aligned} L(x_j, x_i) &= (|1 - 4|^2 + |1 - 5|^2)^{\frac{1}{2}} \\ &= (9 + 16)^{\frac{1}{2}} \\ &= 5 \end{aligned}$$

What KNN algorithm really does is to measure the distance between x_i and all other instances, keeping the k instances with the minimum distances and then returning the class to which most of the k instances belong. Depending on which method we choose to calculate the distance between examples the classification of a new instance may change.

There are some problems working with KNN algorithms when the state space is represented as a Cartesian plane. First, we have curse of dimensionality problems, such as irrelevant features, that increases the distance between instances and the size of the state space. Second, we have scalability problems, because for each new instance we want to classify, we have to measure the distance between all other instances and the new instance. That makes the KNN algorithm complexity increase linearly the time to classify a new instance as the number of available instances grows. Third, we need to set the k value. If k is too small, the algorithm will probably overfit the results, because it will look at a small portion of instances. If k is too large, the results will probably be underfitted, because it will look at too many instances.

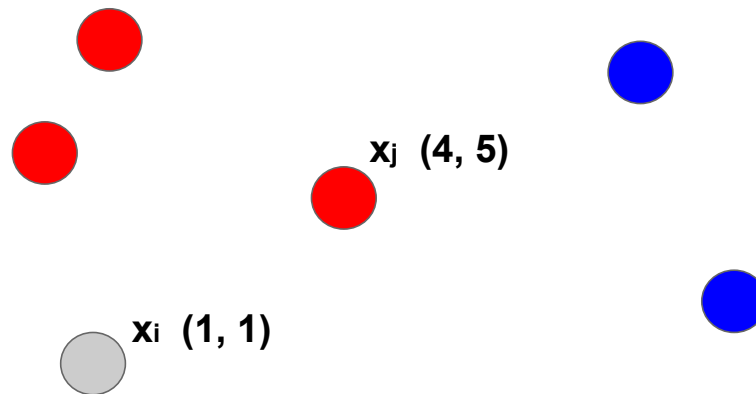


Figure 3.2 – Estimating the class of the new example (gray) by calculating its distance from the other examples

We have seen that machine learning algorithms learn a function that separates examples of each class, but KNN is an unusual classifier because it does not learn a function that differentiates between examples of different classes. Therefore, there is no training phase when using this algorithm. We call it a lazy or instance based algorithm: it just makes estimations when a new instance must be classified, looking in the entire dataset for the k instances that are closest to the new instance.

3.4.2 Support vector machine

Support Vector Machine (SVM) [Vap00] is a binary classifier machine learning algorithm. It represents the input space as a Cartesian plane, and splits the state space in two: one side contains the positive instances, that we label as 1, and the other side contains the negative instances, that we label as -1 .

Figure 3.3 shows a two-dimensional state space with the negative instances in red and the positive instances in blue. SVM splits the state space in two trying to divide the positive and negative instances linearly. It calculates a line in the state space such that all positive instances are on one side and the negative are on the other, creating two boundaries. The black line separates the state space in two, while the red line is the boundary that separates the negatives examples from the rest of the Cartesian plane, and the blue line is the boundary that separates the positive examples.

Consequently, the first problem we want to solve with SVM in the example above is to calculate a straight line that splits the state space in two in order to separate the positive and negative examples. Secondly, we want to calculate the two boundaries that are nearest to the first examples that appear in each side. We need to find a line if the state-space is two-dimensional. If

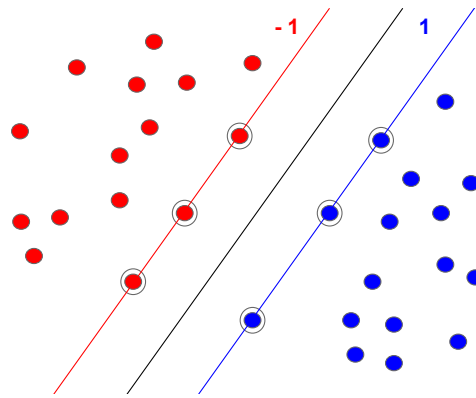


Figure 3.3 – Data linearly separable, with the support vectors inside circles. The separating lines are under the support vectors showing the boundaries between the two classes.

the state-space is three-dimensional we need a plane. And if the state-space is higher-dimensional we need a hyperplane, that is a generalization of a plane for n -dimensional spaces. More generally, we need to find a hyperplane that divides the state space in two, independently of the number of dimensions. There are many hyperplanes that separate the state space, but we want to find the one that separates best. That is, the hyperplane with the largest margins between the positive and negative examples, which we call the maximum marginal hyperplane.

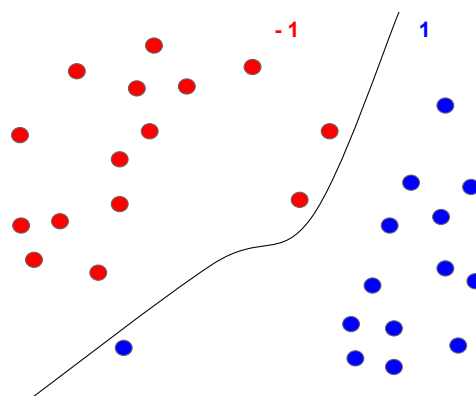


Figure 3.4 – Non-linear separable data, separated by the black line

SVM uses the instances located in the boundaries to calculate the lines that separates the Cartesian plane which we call support vectors. In Figure 3.3, support vectors are represented by the colored circles within black circles under the separating lines. We can calculate these boundaries because we suppose there is a gap between the positive and negative examples. The larger the gap, more accurately SVM can classify new instances because the differences between positive and negative examples become more clear.

The hyperplane in Figure 3.3 is linear because the instances can be linearly separable, then we can use a linear SVM. But some state spaces separates better positive and negative instances if the hyperplane is not linear, for example Figure 3.4, where we cannot draw a straight line to separate them. In these cases we use a variant of the SVM algorithm, the non linear SVM [BGV92].

We also can modify SVM to accept n classes, building a multiclass SVM: we train n SVM classifiers, one for each class, and when we have a new instance, we test it in all classifiers. All of them will return that the new instance does not belong to the algorithm's class, except one, that is the classifier that has the class that the new instance belongs to. This modifications allows SVM classifiers to have the same power as the other classifiers that accepts n classes.

3.4.3 Naive bayes

Naive Bayes (NB) [Mit97] is a probabilistic classifier that uses the Bayes theorem. We explain the Bayes theorem and then move on to defining the NB classifier and the gaussian naive Bayes classifier (GNB).

Bayes theorem calculates the probability some event a is true given that other event b is true, we express the probability of a given b as $P(a|b)$, and the Bayes theorem is given by the formula:

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)}$$

To illustrate the Bayes theorem, consider that 5% of all emails are spam, and the chance an email contains the word *test* given that it is spam is 30%, but the probability of an email contains the word *test* is 10%, independent of being spam. The scenario can be summarized in the following probabilities: $P(spam) = 0.05$, $P(test|spam) = 0.3$, $P(email) = 0.1$. If we want to calculate the chance of an email being spam given it contains the word *test*, we use the Bayes theorem

$$\begin{aligned} P(spam|test) &= \frac{P(test|spam)P(spam)}{P(test)} \\ &= \frac{0.3*0.05}{0.1} \\ &= 0.15 \end{aligned}$$

We can measure the chance of an email being spam given it contains the word *test* and other words, for example *email*, calculating $P(spam|test \wedge email \wedge word1 \wedge word2, \dots)$, expressed as:

$$P(X, Y|Z) = P(X|Z)P(Y|Z)$$

test and *email* are independent words, but both are caused by spam emails, in other words, the the words of an spam email are dependent of the spam, but this words are not dependent between them. Then, we say that *test* and *email* are independent variables, and they are conditionally independent of the variable spam. In the context of feature vectors, we can say that all features in the feature vector are independent, that means each feature contribute independently for an instance classification.

Given the definition of the Bayes theorem, we now turn to the formalization of the NB, a machine learning algorithm for classification using probability. It determines the probability that an instance x , or a feature vector, belongs to a class c , that we write as $c(x)$. We describe each feature of x as $f(x)$, where $f_n(x)$ is the n th feature of x :

$$P(c(x)|f_1(x), f_2(x)\dots f_n(x))$$

In the email example, each feature of x defines if a given email contains a word, *test* and *email* are features in x . It is assumed that each feature of x is independent of its classification $C(x)$ and of other features, we say that x is conditionally independent of $C(x)$. To classify a new instance, the algorithm calculates the probability of x belongs to each class $c_j \in C$, and returns the most probable class that x belongs to, called c_{MAP} . Using the features of x , c_{MAP} is calculated by:

$$c_{MAP} = \arg \max_{c_j(x) \in C} P(c_j(x)|f_1(x), f_2(x)\dots f_n(x))$$

We can rewrite this expression using Bayes theorem:

$$\begin{aligned} c_{MAP} &= \arg \max_{c_j \in C} \frac{P(f_1(x), f_2(x)\dots f_n(x)|c_j(x))P(c_j(x))}{P(f_1(x), f_2(x)\dots f_n(x))} \\ &= \arg \max_{c_j \in C} P(f_1(x), f_2(x)\dots f_n(x)|c_j(x))P(c_j(x)) \end{aligned}$$

Based on the training data, it is easy to estimate each $P(c_j(x))$ by counting the frequency it appears. But estimating $P(f_1(x), f_2(x)\dots f_n(x)|c_j(x))$ is not feasible, because the number of these terms is equal to the number of possible instances (the number of features) times the number of possible target values ($c_j(x)$). To solve this issue, the NB assumption is that the attribute values are conditionally independent of the given target value. That means the probability of $f_1(x), f_2(x)\dots f_n(x)$ is just the product of the probabilities for the individual features, $P(f_1(x), f_2(x)\dots f_n(x)|c_j(x)) = \prod_i P(f_i(x)|c_j(x))$. Substituting this into equation, we have the NB: [Mit97]

$$cNB = \arg \max_{c_j(x) \in C} P(c_j(x)) \prod_i P(f_i(x)|c_j(x))$$

To apply the email classifier example in the NB, consider the training examples in the Table 3.1, that shows in each row if an email is spam or ham and if it contains the words *test* and *email*. The column Email has the values *spam*, *ham* for different emails, and is to be predicted based on the other columns, that shows if a word appears in a given email. To calculate the prediction if a new instance x that contains the words *test* and *email* is spam, we use the NB formula

$$cNB = \arg \max_{c_j \in \text{spam, ham}} P(c_j(x)) \prod_i P(f_i(x)|c_j(x))$$

$$cNB = \arg \max_{c_j \in \text{spam, ham}} P(c_j(x))P(\text{test}|c_j(x))P(\text{email}|c_j(x))$$

To calculate cNB, the formula requires some probabilities that can be estimated from the training data.

$$P(spam) = 3/5 = .6$$

$$P(ham) = 2/5 = .4$$

$$P(test|spam) = 3/3 = 1$$

$$P(test|ham) = 1/2 = .5$$

$$P(email|spam) = 1/3 = .33$$

$$P(email|ham) = 1/2 = .5$$

With this probabilities, we can estimate the class of the new instance

$$P(spam)P(test|spam)P(email|spam) = .198$$

$$P(ham)P(test|ham)P(email|ham) = .1$$

The NB classifier returns *spam* for the new instance, based on the estimations calculated from the training data.

<i>test</i>	<i>email</i>	Email
1	1	spam
0	0	ham
1	0	spam
1	0	spam
0	1	ham

Table 3.1 – Training examples for email classifier

Gaussian Naive Bayes [Mit97] is a variation of NB that assumes the likelihood of a feature is Gaussian. Then, in order to build a GNB classifier, we must calculate the mean (μ) and the variance (σ^2) of each class in the dataset and apply the normal distribution formula.

4. FEATURE SELECTION FOR FMRI DATA

In Section 3.3.1, we discuss how to create generic examples of any domain for classification. We viewed how to reduce dimensionality of any type of data for increasing classification accuracy using feature selection algorithms, which can be implemented in a number of ways. Because we work with a very specific type of data, instead of describing traditional feature selections algorithms that are useful for various domains, we work with feature selection algorithms widely used with fMRI data and classification.

The data of each patient is composed of over 30,000 voxels, most of which are not involved in the neural activity regarding the pseudo word task performed by patients from ACERTA. Since such irrelevant voxels do have activation values, they can interfere with the training of the classifier. Consequently, we want to remove irrelevant voxels from the dataset before using the classifier in order to generate cleaner results. Thus, instead of using all of the 30,000 voxels, we use feature selection methods in order to transform the fMRI data into relevant features for the classifier. All the feature selection methods we use in the experiments in Chapter 6 are detailed below.

4.1 Most Stable Voxels

In the first feature selection approach we evaluate what was proposed by Buchweitz's et al. [BSM⁺12] by selecting a fixed number of *most stable voxels* [PMB09]. By stable voxel we mean a voxel that has a minimal standard deviation value for its activation over the times when patients are seeing words within the time series. We describe this technique in further detail in Section 5.3. This means that these voxels are consistently activated throughout the tasks. Following this approach, this method consists of selecting voxels to be more or less evenly distributed throughout the brain instead of being clustered in just a few brain locations (otherwise, activation tends to cluster around the occipital lobe, due to the nature of a visual task). Therefore, we partition the brain into 4 lobes (occipital, temporal, parietal, frontal) and find the n most stable voxels in each lobe, resulting in $n * 4$ most voxels distributed over the brain which will be used as features for the classification algorithm.

Although the number of chosen voxels is arbitrary, classifier accuracy does not increase much as we increase the number of chosen voxels. For example, Buchweitz [BSM⁺12] et al. argues there is no need for choosing more than 2000 voxels, and in that case, 120 voxels were enough for the classifier to perform above chance.

4.2 Region of Interest

A Region of Interest (ROI) in the context of neuroimaging is a defined portion of the brain that is important for the study at hand. We separate a ROI from the rest of the brain by defining the voxels that belong to the ROI. For example, Figure 4.1 shows voxels selected from the inferior frontal region of the brain. This ROI is adequate for neuroimaging studies about reading, as in most people this brain region is known to be used for reading.

fMRI studies hypothesize that the voxels from a specific region have a different pattern of activation from the other brain areas when patients are performing a task. Thus, by separating the brain in ROIs, we can answer various questions about our study. If we are investigating the brain anatomy, we can use the voxels of specific ROIs to answer questions such as 'is region x involved in task a ?' or 'can a classifier using only voxels from region x differentiate between tasks a and b ?' [EGK09].

In order to separate the voxels from one ROI to the rest of the brain, we need to define where that ROI is located. There are several approaches to define a ROI. We can manually extract the voxels that belong to a ROI in the data of each patient, setting the boundaries of the voxels that belong to an ROI. This method is very precise, because the brain anatomy varies and the same ROI can be in slightly different locations in each patient. One drawback of this method is that we need a brain anatomy specialist to define the ROIs. Nevertheless, there are easier ways of separating the ROIs, such as using atlases that calculate the location of a ROI in a single subject based on statistical brain maps [FHW⁺94], or using masks that already define where each brain region is located, such as the masks in Figure 4.2, which divides the brain in anatomical and functional regions.

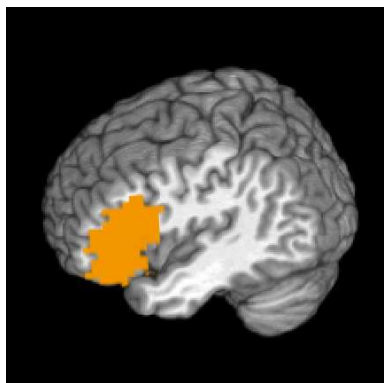


Figure 4.1 – Example of a Region of Interest. This ROI covers all voxels from the inferior frontal part of the brain, which is involved in the language production task. Thus, it is appropriate for neuroimaging studies about reading, as ACERTA project.

After defining the ROI, we use it to generate features for the classifier examples. The simplest way of extracting features is to use all voxels from the ROI and convert each voxel in a feature. Because classifiers often need no more than 500 voxels for achieving reliable accuracy, if the ROI is large and contains many voxels, we can choose only a subset of them to use in classification [EGK09]. Further, because each patient has a different head size, not all patients data have all

the voxels in an ROI, specially if that ROI is located on the edges of the brain (although some preprocessing, as spatial normalization, can fix that problem by making all the patients head of the same size). For this reason, choosing a fixed number of voxels in a ROI is reasonable when we want all feature vectors to have the same number of features, so we can compare the data of different patients.

The last step of working with ROIs is to prove the classification result is reliable. For that reason, we use the same classifier with features of a second ROI that should not be involved in the task. If the first ROI provides a better classification than the ROI not involved in the task, we say it is unlikely that the voxels from a region not involved in the task provide a better classification than voxels involved in the task, proving the significance of the classifier result [EGK09]. For example, when using ACERTA data, in which children perform a written word task, we expect the classification using voxels from inferior frontal gyrus (related to language production) to have better accuracy than using the voxels from auditory cortex, since the only sound children hear during scanning is the noise of the MRI machine.

4.3 Parcellations

Parcellations are brain masks that define how the voxels in the brain are divided into smaller regions. For example, Figure 4.2 shows 3 different parcellations and how they divide brain regions into smaller parts (AAL, cc200 and cc400). To generate the activation of a single parcellation, we calculate the average activation of all voxels inside the parcellation. The problem of measuring the activation of individual voxels is that they are noisy and often do not represent the main activation of one specific brain region. Thus, we expect that increasing the feature selection granularity using the average activation of many voxels results in an increase in the reliability of the resulting examples, although we lose most of the information that the voxels in that brain portion contains.

We use this feature selection method when we want to study the whole brain and the interaction between brain regions. For example, we can answer questions such as "can we find the brain regions that integrate the neural network involved in task a ?" or "can we find the brain regions that make the classifier differentiate between tasks a and b ?"

There are important differences between ROI and parcellations feature selection, although both can use the same mask for defining brain regions or ROIs. When using ROI feature selection, we have an a priori hypothesis about the brain functions involving a task, and test only that region in detail. Conversely, parcellation feature selection does not assume which brain areas are involved in the task, summarising (calculating the mean of all voxels) and testing all the brain areas.

In this work, we test 3 different masks, and each mask contains its own parcellations definitions. The first mask is the AAL mask [AGMGVH06], which divides the brain into 116 anatomical regions (first line of Figure 4.2). From these regions, we remove the ROIs that belong to the cerebellum (which is known to have no important activation for cognitive tasks, and is not scanned

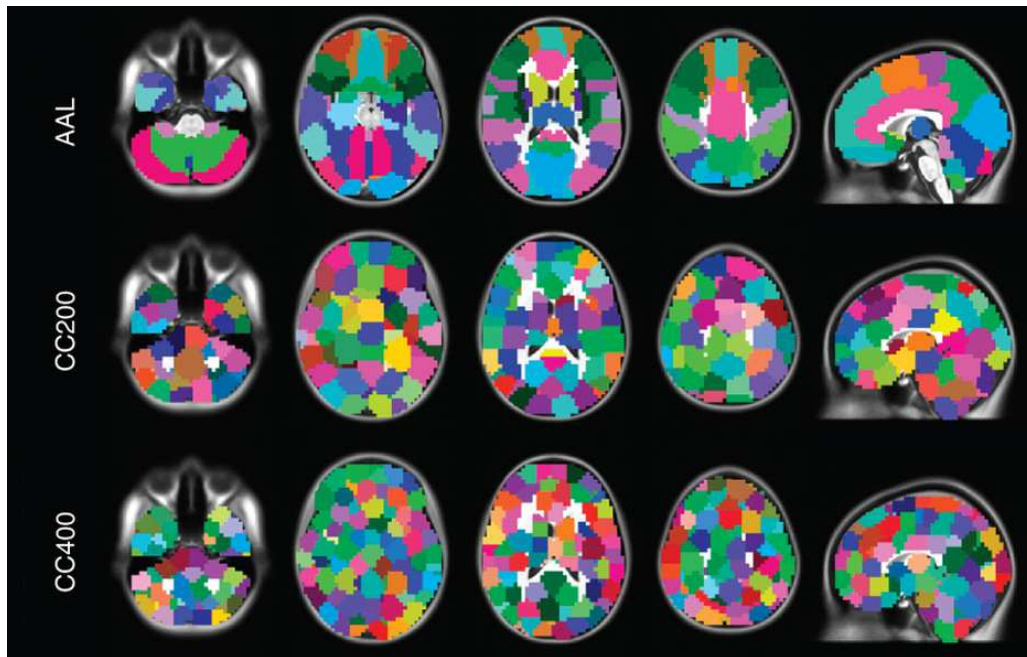


Figure 4.2 – 3 types of parcellations: AAL divides the brain in 116 anatomical regions, cc200 divides the brain in 200 functional regions and cc400 divides the brain in 400 functional regions. From Craddock et al. [CJH⁺12]

in the scan session), obtaining 90 parcellations for the experiments. The second mask is cc200 [CJH⁺12] (second line of Figure 4.2). cc200 mask contains 200 regions divided functionally. That means the voxels are clustered by activation similarity, differently from AAL, which is parcellated by anatomical similarity. The third mask is cc400 [CJH⁺12] (third line of Figure 4.2), which group voxels by similarity as cc200, but divides the brain in more parcellations (400). The authors of cc200 and cc400 recommend using cc200 when we want more readability in the results, as there fewer brain areas to interpret in this mask. Conversely, when we want more accuracy in our results, cc400 is recommended [CJH⁺12].

4.4 ReHo

Regional Homogeneity (ReHo) [ZJL⁺04] is a feature selection algorithm commonly used with rsfMRI data [LLL⁺06]. It measures brain activity by calculating the similarity between a single voxel and its nearest neighbors. This method is based on the hypothesis that brain activity is concentrated in voxel clusters instead of just a single voxel.

Applying ReHo to rsfMRI is appropriate because it makes no prior assumption of how the BOLD signal is. As we have seen in Section 2.2, in task-based fMRI we know what subjects are doing inside the MRI scanner, so we are able to model the hemodynamic response related to the task, while in rsfMRI, we do not know what subjects are doing, then we cannot model the BOLD signal (which means guessing in what time the BOLD signal will be higher and lower, as we show in Figure 2.4). By not supposing how the BOLD signal is, ReHo can gather more information than

methods that requires a predefined model of the BOLD signal. Because we cannot predict what subjects are doing inside MRI scanner, then we cannot predict the resulting BOLD signal. As a consequence, ReHo identifies unexpected BOLD signal patterns, because we are not looking for any pattern in particular. Although this algorithm is designed for rsfMRI, it can be used for task-related fMRI data when using block design or slow event related paradigms.

To calculate similarity between voxels, we use Kendall's coefficient concordance (KCC) [Ken90]. KCC is a statistical test that compares similarity between any number of group of voxels. Its values are in a range between 0 and 1, where 1 means all voxels in a group are equal, indicating great similarity. KCC is different from the t-test method described in Section 2 because it compares any number of voxel clusters and does not assume data has a normal distribution.

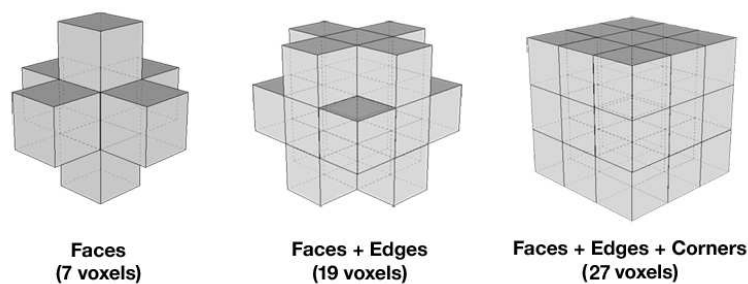


Figure 4.3 – The shape of the cluster ReHo algorithm analyzes to find clusters with high similarity. Each cube is a voxel. from C-PAC [SCK⁺14]

The number of voxels ReHo algorithm cluster can be set as 7, 19 and 27, as shown in Figure ???. That means the algorithm looks for clusters of voxels of the determined size that have a similar activation. Figure 4.3 shows the different results we can obtain by choosing the cluster size. As we increase the number of contiguous voxels we analyse the similarity, less areas with a higher KCC we get.

4.5 Whole Brain

This is the simplest feature selection we use in this work. It consists of extracting from the fMRI data all voxels that are inside the brain with a binary mask (Figure 4.4) and use them as features. This set up results in examples with about 30,000 features, which are approximately the number of voxels inside the brain. The advantage of this technique is that it allows the classifier to look for patterns that are spatially distributed across the brain instead of using feature selection methods that assumes a predetermined hypothesis about where those patterns are. Although using all the voxels of the brain from classification seems to produce noisy examples, it can outperforms other feature selection methods, such as getting only voxels from an ROI [HMB⁺11].

We can simply use raw voxels as features, but there are many ways of extracting more information of each voxel by transforming each one in multiple features. For example, instead of extracting all the voxel time series, we can calculate the maximum voxel time series, the average



Figure 4.4 – Binary mask used for extracting all the voxels that are inside the brain.

time series and maximum and minimum correlation coefficient between the voxel and its neighbours [SC14].

4.6 ANOVA

Analysis of Variance (ANOVA) is a standard approach when analysing fMRI data, and is frequently used with classification for reducing the number of voxels [SEVA09] [ABMSP12]. It is a statistical method for separating groups that are mixed, which compares the means of two groups to ensure they were generated by different sources. ANOVA is a reliable method because it performs multiple statistical tests to validate the data can be split in groups. When using ANOVA with fMRI data, we say each group is a condition (e.g. words and pseudo words are different conditions). For example, a voxels that always have variance equal to 2 when a patient is reading a word, and always have variance 20 when a patient is reading a pseudo-word probably can help the classifier to distinguish what type of word the patient is reading. This method is adequate for fMRI data because different conditions are supposed to generate different BOLD signal, and consequently, generate distinct variances.

When studying ANOVA In more detail, we notice it returns a statistical measure called f-score for each voxel, which indicates the level of variance of that voxels between groups. A common approach for choosing voxels with this method is to get the voxels with higher f-score [HHS⁺09]. For example, in this work we use as features the 5% voxels with higher f-score.

5. RELATED WORK

We have surveyed related work on the application of machine learning to fMRI data and have found 4 techniques we consider to be related to our work. In this chapter, we study these techniques and compare them with our work. The first application tries to improve a classifier error rate by developing two new feature selection methods and comparing these methods with two well known feature selection methods. This analysis is performed on the data of patients with major depressive disorder [CHHM09].

The key similarity between this work to ours is that they use clinical subjects to identify a neurocognitive disorder. In the same way, they try many feature selection algorithms to compare the final SVM accuracy using each one. Finally, they use a connectivity map as the feature vector, which is an experiment we try with our data. The second application predicts which children with developmental dyslexia improves their reading skills after a certain number of years using the support vector machine classifier [HMB⁺11]. Both works use data from children with dyslexia to train support vector classifiers. Thus, we expect to obtain similar results, especially the most important cortical locations for the classifier. The third describes a feature selection single-subject method we use in the experiments [BSM⁺12]. The feature selection consists of choosing just a few number of voxels from a task fMRI data in order to train a GNB. Finally, the last reports an fMRI study on the differences between good and poor early readers [PTP⁺01]. This work emphasizes how fMRI data analysis and behaviour measures correlate, and gives a description about current finding about reading difficulty neural basis. We expect to find the same cortical locations hiper and hipo activated in our data.

5.1 Improving classifiers accuracy by using different feature selection methods

Craddock et al. [CHHM09] propose a technique to detect major depressive disorder using support vector machine classifier. In this study, the resting state fMRI (rsfMRI) data of 20 healthy subjects and 20 patients with major depressive disorder and no other clinical disorder was acquired in order to use different feature selection methods and a machine learning methods to differentiate between patients and controls. This work develops two new feature selection methods and compares these methods with two state-of-the-art methods. The final classifier accuracy varies according to the feature selection method, demonstrating that choosing the right feature selection for the data is critical for the resulting classifier accuracy.

In order to prepare data for the feature selection algorithm and the machine learning algorithm, functional connectivity (FC) maps are created for each subject. FC maps show which brain areas communicate with which other areas, highlighting the correlations between spatially remote brain activity. An example of FC is figure 5.1, which shows a matrix where rows and

columns are ROIs of the brain, demonstrating brain areas that have different connectivity patterns between groups, depressed vs. controls.

If two areas have the same connectivity pattern, we say that they are correlated (yellow squares).

Conversely, they have different connectivity patterns, we say they are anti correlated (blue squares). Differences between brain networks of controls and patients serve as markers for diseases.

State-of-the-art resting state FC (rsFC) analysis is divided in two parts. First, a subject-specific FC map is generated using univariate methods, such as correlation analysis and ICA that we have seen in Section 2.2.2. Second, a second level statistical analysis (t-test) compares the FC map between groups. Because of the drawbacks of univariate methods, the authors propose to analyse rsFC with multivoxel pattern analysis (MVPA). MVPA is a classifier that uses at all voxels at the same time, in contrast to univariate methods, that use just one voxel at time. A SVM is used in this application, as it is sensitive to spatially distributed patterns of FC and is less sensitive to noise. Four feature selection methods are used in order to increase the SVM accuracy, where a feature is an FC value between two brain regions (one cell of the rsFC matrix). In this context, each brain region is represented by a voxel chosen by a brain anatomy specialist.

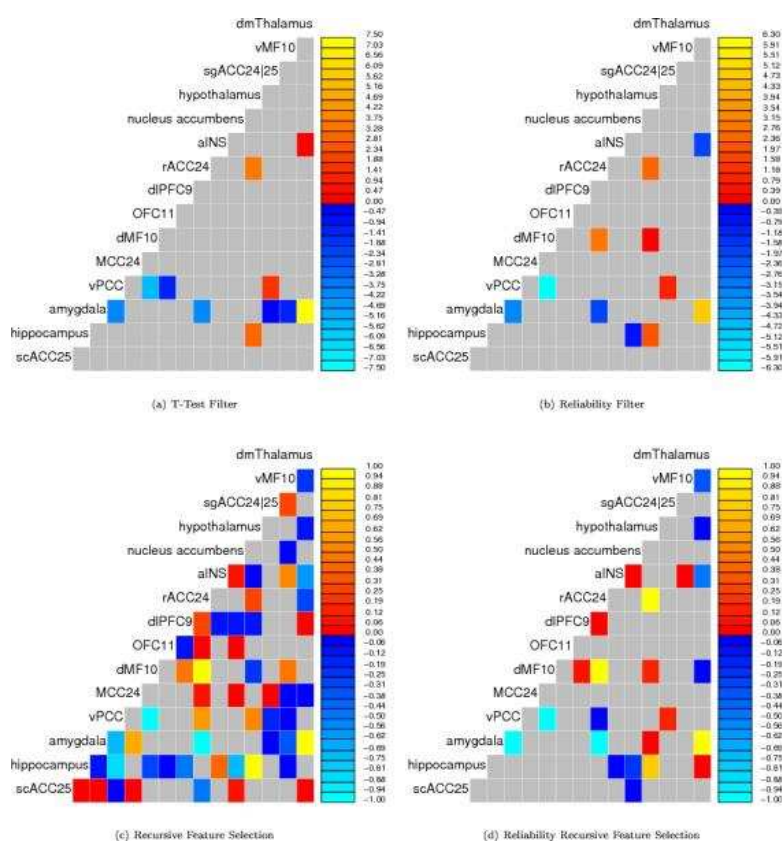


Figure 5.1 – From [CHHM09]. Connectivity matrix showing how 15 brain regions communicate. The connectivity matrix varies according to the feature selection method. Yellow squares show positive correlation between two areas while blue squares show negative correlation. Gray squares show no correlation between areas.

The purpose of the study is to apply two new feature selection methods in order to improve SVM accuracy when using rsfMRI data. A filter method (reliability filter) and a wrapper method (reliability reverse feature elimination) are created and compared to standard feature selection methods (TF and RFE) (see Section 3.3.1).

- TF (t-test filter) is a filter method that uses t-test to determine which features differentiates between two groups, eliminating features with lower scores.
- RF (reliability filter) is a filter method that uses bootstrap methods (see Section 3.3.2) to split the data between training set and test set. It uses the test set to train an SVM and selects features that are chosen as support vectors. Consequently, RF selects a set of features that really make sense for machine learning algorithms, because it choose the features that separate the data between two classes (the features that are support vectors are in the edge of the hyperplane that splits the cartesian plane in 2, dividing the examples of each class).
- RFE (reliability reverse feature elimination) is a wrapper method that is used iteratively with bootstrap methods. It ranks features according to the final classifier accuracy using that set of features in each iteration and excludes 10% of the features with lowest ranks.
- RRFE (reliability reverse feature elimination) is similar to the RFE method, but it calculates the reliability of features in order to select the ones that will be removed in each iteration.

SVM distinguished better between controls and patients using the RF feature selection method, with 95% accuracy. With no feature selection, SVM accuracy dropped to 62.5%, while state-of-the-art methods are not able to distinguish between patients and controls. Figure 5.1 shows the matrix generated by each feature selection described above. We want to highlight that as this matrix represents a few brain regions, it is not just computer readable. Anyone with brain anatomy knowledge can interpret it and have insights about the cognitive disorder of the study. Additionally, the classifier accuracy can give us a deeper understanding of which feature selection method generated the most reliable connectivity map.

5.2 Predicting Dyslexia

The work by Hoeft et al. [HMB⁺11] investigates the brain mechanism that children with dyslexia use to improve reading skills using SVM classifiers. Some children and adults with dyslexia can compensate their poor learning abilities and achieve reasonable reading scores over time, although they do not reach standard scores. It is unknown why some people with dyslexia can develop this compensation while others others cannot, as well as the brain parts involved in that compensatory mechanism. No behavioral measure (standard reading and writing tests) is able to predict improvement in reading skills of dyslexic children above chance, that's why studying the neural basis of this compensation can lead to a better understanding of how this mechanism works.

Studies on patients with dyslexia using fMRI and reading tasks reported hypoactivation in the left parietotemporal and occipitotemporal regions, and hyperactivation in left and right inferior frontal gyri (IFG) [PTP⁺01] [SMvdM⁺08]. This hyperactivation may indicate the compensatory mechanism children with dyslexia develop in order to improve their poor reading skills, as they are not found in controls and they grow over time in children with dyslexia.

In order to understand such compensatory mechanism that improves over time, 25 children with dyslexia and 20 healthy controls performed three types of tests. First, they performed a written rhyme word task in an fMRI experiment. Second, subjects underwent to Diffusion Tensor Imaging scans. Finally, they took other tests to provide reading behavioral measures. Two and a half years later, the behavioral tests were taken again to measure improvements on the children reading skills. Two analyses were made with the SVM classifier to predict if a child with dyslexia would improve its reading scores after 2.5 years: univariate analysis and multivariate pattern analysis (MVPA). Both used the fMRI data taken when children were performing the rhyme task at the beginning of the experiment.

For the univariate analysis, two possible compensatory areas for children with dyslexia were used, the left and right IFG. The results show that the behavioral measure of a single word reading correlated positively with the right IFG activation. Moreover, when using the data of the whole brain the only positive correlation found was with the right IFG for children with dyslexia; the same correlation was not found in the control group.

For the MVPA analysis, children with dyslexia were separated into two groups: one in which children increased the single word reading measure after 2.5 years, and another the group where children did not improve. A whole-brain MVPA was performed using the voxel intensity of the contrast image (i.e. signal change between rhyme task and resting state) with a linear SVM classifier using leave one out cross-validation. The results show 92% accuracy when classifying whether a single child would improve its reading skills or not. When using the two ROIs (right and left IFG) instead of the whole brain, the accuracy decreases to 72%. Further, the classifier indicates that the voxels that contributed more to the classification are located in the right IFG, left prefrontal cortex and left parietotemporal region. Finally, the results reported that even though the classifier accuracy is high when using only fMRI data, adding behavioral measures to the examples leads to a decrease in the classifier accuracy.

The authors conclude the univariate and MVPA results show evidence that the right IFG exhibits a greater activation while children with dyslexia perform the rhyme task, and this hyper activation correlates positively with reading scores. Therefore, we can say that the right IFG plays an important role in the improvement of reading skills specifically for children with dyslexia, as these pattern activations and correlations were not found in the controls. Furthermore, the results show the MVPA analysis predicted improvement in reading skills much better than the behavioral measures, and the analyses with the whole brain are more precise than using just specific ROIs.

5.3 Most stable voxels feature selection

The work of Buchweitz et al. [BSM⁺12] aims to understand brain pattern activations associated with the semantic representation of nouns in bilinguals (speakers of portuguese and english). More specifically, the authors investigated whether the brain pattern activation for one noun in one language is similar to the pattern activation for the same noun in another language. They hypothesize that if a classifier can identify the word a subject is reading when trained in one language (L1) and tested in other (L2), then the semantic representation of the word is independent from language.

For this purpose, participants that are natural Portuguese (L1) speakers and learned English (L2) later were chosen for the study. They were instructed to read 14 concrete nouns in Portuguese in one scan session and the same nouns in English in a separate scan session. Each scan session is separated in 6 blocks, in which the 14 words are presented in random order. Thus, a word is presented 6 times in L1 and 6 times in L2. The duration of the stimulus is 3 seconds, followed by 7 seconds rest.

A classifier was trained in one language and then tested in other language. We highlight 3 key points used to generate the examples. First, as there is a huge amount of voxels in the scan data, feature selection was performed. The 120 most stable voxels of each patient were chosen in order to create the examples. In this context, stable voxel means a voxels that have a minimal standard deviation while participants are seeing a word. Second, while creating one example per word, the authors used the average percent signal change of the chosen voxels when participants see that word. Then, they skip the first 4 images when a participant started to see a word, and use average of the next 4 images to create an example. Notice that skipping some images and taking the average of imgages taken some seconds later is a reliable method for using only the images where the BOLD signal should be activated, as we have seen in Section 2.2. Finally, the examples are normalized, so the average is 0 and the standard deviation in 1. Normalizing the examples make all features have the same importance.

For the classification, a gaussian naive bayes method is used, and the examples are divided in 14 classes (one class for each noun). As there are many classes in this experiment, rank-accuracy is used to measure the prediction accuracy of the classifier. That mean instead of the classifier predicts a class that an example belong, it return the probability of the example being of eahc class, creating a rank indicating which class is the most probable for the example. Two classifiers are trained for each participant, the within language classifier (the classifier is trained and tested in the same language) and the across language classifier (the classifier is trained and tested in the same language). When measuring the accuracy of the within language classifier, k-fold cross validation was used, leaving 2 examples out from the 6 examples of each word. The average accuracy of each fold was calculated, and the final classifiers accuracy is the average accuracy of all folds. For the across language classifier, all the examples in one language were used as the training set while the examples of the other language was used as the test set.

In the across language classification, the classifier was able to identify which of the 14 words the participant was reading. When training the classifier in L2 and testing in L1, the classifier accuracy is 68%, while when training in L1 and testing in L2, the classifier accuracy increases to 72%. The most stable voxels chosen for this classifier are located mainly in the left hemisphere of the brain, more precisely in the frontal and occipital lobe. Additionally, an overlap was found in between the most stable voxels of L1 and L2. For within language classification, the classifier accuracy in L1 and L2 are 60%. A correlation between the classifier accuracy of the across language and within language classification was found within subjects.

5.4 Dyslexia network

Most children with reading disabilities have problems with phonological awareness (PA), which is the knowledge that spoken words are formed by smaller sounds (syllables and phonemes). PA can be measured by phoneme deletion and blending tasks, and has a strong correlation with word reading outcome in early readers (this task can predict the future reading measure of children). Consequently, there is a strong relation between PA and reading acquisition, poor readers often have low scores on those tests. PA helps in the development of visual word recognition for two reasons. First, PA helps in the development of visual word recognition, as it prepares children to recognize the small pieces of spoken words. Second, PA supports the association of visual representation of words (grapheme) with the phoneme they represent, that is the key skill for reading. Grapheme to phoneme conversion is measured by pseudoword reading task, that is strongly correlated with PA and is a predictor of word reading outcome. Thus, Poldrack et al. [PTP⁺01] tries to correlate those measures with fMRI activation patterns in early readers.

fMRI studies often show the differences between typical readers and poor readers in the left hemisphere regions, in which the language network is located in typical readers. More specifically, those differences are located in the left hemisphere posterior areas, where poor readers show under-activation in temporo-parietal and occipitotemporal locations. Further, poor readers show evidence of a compensatory mechanism to their poor skills by hyperactivating the posterior regions of the right hemisphere and in the left and right frontal lobe.

Besides the language processing impairments, the authors also investigate if the PA deficit can be explained by visual motion processing and auditory processing, what could increase the complexity of the well known reading network. Additionally, a brain region that plays an important role in early reading and could differentiate reading difficulty (RD) and traditional development (TD) is the thalamus, which is involved in learning tasks. In summary, the authors hypothesize the early reading network is wider than we expect and that this network can differentiate between TD and RD.

For this purpose a fMRI study with 62 English speaking children was performed, the subjects were beginner readers with and without RD. Behavioral measures of reading and listening

were also taken. The fMRI paradigm consists of a match/mismatch judgment where subjects viewed a picture and read or listened a word or a pseudoword and judged if the picture and the word match.

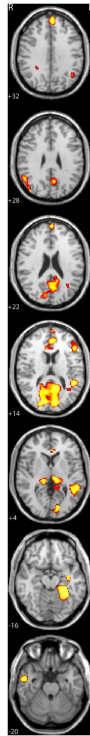


Figure 5.2 – From [PTP⁺01]. Brain areas activated during the print word stimulus that correlate with writing behavioral measures. That means poor readers activate less the highlighted areas than good readers. The left hemisphere is at the right side of the image.

When analysing the fMRI data, the authors reported that while reading printed words, the most important activated brain regions are: left superior temporal gyrus, left fusiform gyrus and left thalamus. The activation of these regions are correlated with behavioral measures and are shown in Figure 5.2. Moreover, while listening to spoken words, the most activated brain areas are: left inferior frontal gyrus, left precuneus and posterior cingulate gyrus. A smaller activation was found in the right superior temporal gyrus, thalamus and fusiform gyrus.

The authors discuss that the neural reading network of skilled early readers is wider than the network of RD readers. That means skilled readers recruit more brain areas than poor readers. This network is composed by the brain regions described above. A positive correlation was found between behavioral speaking and listening scores and the activation of the following regions: left temporoparietal and occipitotemporal regions, left inferiorfrontal gyrus, precuneus, posterior thalamus, prefrontal cortex and right parietal and temporal networks. The reading neural network in early readers is more distributed than in proficient late readers. Because early readers need more support for the reading task, more brain regions are recruited. Therefore, as early readers become more skilled, the network becomes more efficient. Additionally, the brain locations recruited for the reading and listening task in the study shows that PA in early readers involves more networks than just the language network.

5.5 Discussion

The works described in this chapter demonstrate how fMRI studies can provide good evidence of how the brain perform daily tasks or to get a deeper understanding of cognitive diseases. The tree first works use machine learning to analyse the fMRI data in different ways. The first one shows that the feature selection methods we choose for the data can change dramatically the results we get for the feature vector (see the differences between the feature vectors generated by the 4 feature selection methods in Figure 5.1) and later from the classifier accuracy. The second uses SVM algorithms to predict reading skills gains in dyslexic children, in a context in which behavioral measures are not enough to do the same task. Besides not predicting reading skills gain, those behavioral measures decrease the classifier accuracy when added to the feature vector. Further, the paper shows the classifier accuracy is maximized when we use the whole brain to do the classification instead of using just some regions of interest. This statement complies with the idea that the brain works with networks rather than individual regions, and we must use classifiers to analyse many brain regions at the same time in order to find those distributed networks. The third work describes a feature selection technique used in our experiments and details how to deal with fMRI data, such as what part of the time series we need to average to generate the examples for the classifier. The last work does not use classifier for the fMRI analysis, instead it discriminates the brain areas involved in the PA in early readers. Additionally, the study shows correlations between some behavioral measures and activation level in some brain regions. In our experiments, we should find some of the brain areas reported in this study, as the authors point those areas are found in many other neuroimaging language studies.

6. EXPERIMENTS AND RESULTS

We now describe the experiments we conducted on fMRI data from ACERTA project. Our ultimate goal is to discover which brain activation patterns children with reading impairments use for reading and discriminating between words and pseudo words. In this sense, we use a Linear Support Vector Machine classifier, detailed in Section 3.4.2, to discover those patterns and test some of the feature selection algorithms from Chapter 4. We aimed to test whether feature selection algorithms can provide a better classification accuracy as well as more reliable results about the brain regions involved in the pseudo word task. We describe how data was collected in the scanning session and preprocessed in Section 6.1, how we generate examples for classification from the pre-processed data in Section 6.2.

We divide our experiments into two types. Initially we test the classification algorithms on the data of single patients in Section 6.3. However, we are interested in the brain patterns children with reading impairments use rather than the brain patterns of individuals. Thus, the second set of experiments use the data of all patients together in Section 6.4, testing the same classification and feature selection techniques to discriminate the brain patterns underlying the reading task, generalizing the classifier to be trained with data from some patients and identifying tasks on other patients. Note that we do not use Reho and Most Stable Voxels methods because we were only able to apply these methods in the single subject experiments. That is because these methods choose voxels using the time series data of a single subject, selecting voxels in different locations in each subject's data. For performing the cross subject classification, we need the feature selection to choose voxels that will work in the exact same brain regions throughout all the subjects. Figure 6.1 shows a schematic view of how the whole experiment was performed: image acquisition, processing, generating cleaner images for the classifier (contrast images, in cross subject experiments) using feature selection and classification. Finally, we discuss the results in Section 6.5.

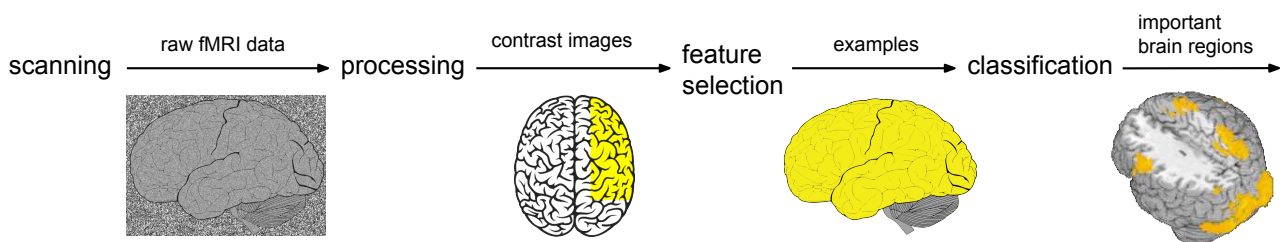


Figure 6.1 – Schematic view of experiments from data acquisition through identifying the important brain regions for reading. First, in the scanning session, children read a list of words while the MRI machine acquires the raw fMRI data. Second, the raw data is processed, cleaning the noisy fMRI data and generating contrast images. Third, the contrast image is used in the feature selection algorithm. In this example, we use the whole brain feature selection, in which all voxels in yellow (that are inside the brain) are chosen. The feature selection algorithm generates examples for classification. Finally, we use the examples with the SVM classifier, which shows us the important regions for reading.

6.1 Data

We briefly describe the data from the ACERTA project we use in this work. Some study parameters reported here are out of the scope of this work and are not detailed further (but see [HSM04]). Nevertheless, the specific parameters must be described in any fMRI study in order to allow reproducibility.

Participants Children with complaints of persistent reading difficulties after two years of formal schooling were selected for the ACERTA project. Participants undergo a psychological and medical evaluation (I.Q. and medical history), and a series of reading and writing evaluations with a speech therapist. After the evaluation, children diagnosed with dyslexia were scanned at the Brain Institute: 10 participants (4 female); mean age 10.2 years ($SD = 1.68$, range = 8 - 13). Right and left handed people has different brain configurations. While most right-handed people has language related brain regions in the left hemisphere, most left-handed people has the same language region in the right hemisphere. Because is easier to analyse fMRI data if all participants have the same brain configuration, specially the language related regions, in this study we only use right-handed children. The present study was approved by the Pontifical Catholic University Research and Ethics Committee (process number 3629513.0.0000.5336). At this moment, we have a number of patients with reading difficulties, but only a few healthy controls. For this reason, in this work we only use data from children with reading difficulties. Nevertheless, in the future, when we acquire the same number of scan of children with reading difficulties and controls, we want to compare the reading brain patterns from the two groups.

Paradigm An event-related experiment was conducted using a word and pseudo word reading task. The set of stimuli is controlled for regularity of letter-sound association, word length (long and short words), and frequency (frequent and infrequent). The reading task contains 20 regular words, 20 irregular words and 20 pseudo words. The 60 stimuli were divided in 2 runs with 30 trials each. Each stimulus was presented on a screen for 7 seconds with the question “is this a real word?”, and the participant had to answer “yes” or “no” by pressing a button. The inter stimulus interval ranged from 1 to 3 seconds. The 2 baseline conditions consisted of the presentation of a white cross in the middle of a black screen for 30 seconds.

Scanning All data was collected on a GE HDxT 3.0T MRI scanner. Patients underwent a T1 structural scan ($TR/TE = 6.16/2.18$ ms, isotropic $1mm^3$ voxels) and then, a two 5min 26sec functional FMRI EPI sequence, which was performed with the following parameters: $TR = 2000$ ms, $TE = 30$ ms, 29 interleaved slices, slice thickness = 3.6mm matrix size = 64x64, $FOV = 216 \times 216 mm^3$, voxel size = $3.4 \times 3.4 \times 3.6 mm^3$.

Preprocessing The data for the single subject and cross subject experiment underwent different preprocessing steps. For the single subject experiment, functional data was upsampled to have a $TR=1$ time resolution. The first 6 seconds of each functional run was discarded to eliminate

T1 equilibrium effects and subsequently concatenated. Data was then despiked, slice-time and motion corrected, blurred with a 6mm full width-half-max Gaussian kernel, and aligned to a standard space (MNI152) using the T1 structural volume for improving the registration. Finally, in order to further remove noise from the data, a general linear model was calculated using the motion estimation parameters as nuisance variables. All preprocessing was performed using the AFNI software.

For the cross subject data, Statistical analysis was performed using the statistical parametric mapping software (SPM8) and images were preprocessed for slice-time correction, realignment, coregistration, normalisation and smoothing. A first level multiple regression was performed with the 3 task conditions (words) and baseline.

6.2 Example generation for fMRI data

When generating examples for the classification experiments, we need to reduce the 640 seconds of data into a format suitable for processing in the specific machine learning approaches we use in this work. For single subject experiments, we use two approaches to transform each of the 60 stimuli into one example, while in the cross subject experiment we use one technique to transform the the time series of a single patient into one example of each class (regular word, irregular word, pseudo word).

6.2.1 Mean 4 Seconds

In the first method for generating examples we follow Buchweitz et al. [BSM⁺12] (discussed in Section 5.3) and other studies [SMM⁺08] [MSC⁺08]. In our study, each word remained on the screen for seven seconds. The brain imaging data used consists of the average activation of four seconds of a voxels for images collected two seconds after the moment each word was presented and the images in the following four seconds; if the stimulus starts in time point 1, we use the time points 3,4,5,6. In this sense, we try to average the activation time points where the BOLD signal is at its maximum, as , as illustrated by the activation graph from Figure 2.4 in Section 2.2

6.2.2 Betas

When using data from an event related paradigm, as we use in our experiments, sometimes the signal of two tasks overlap because they are too close and there is no time for the BOLD signal to return to baseline, Figure 6.2 illustrate how the BOLD signal of two tasks can overlap. Consequently, averaging the BOLD signal in this case is not helpful. For overcoming this problem, we can estimate a single value for each stimulus separately. Because we have 60 stimuli, we get 60 values which

represent the activation in a voxel without interference of near tasks. Instead of each voxel having 640 time points (since scanning session has 640 seconds) we reduce this number to 60 time points, one for each task.

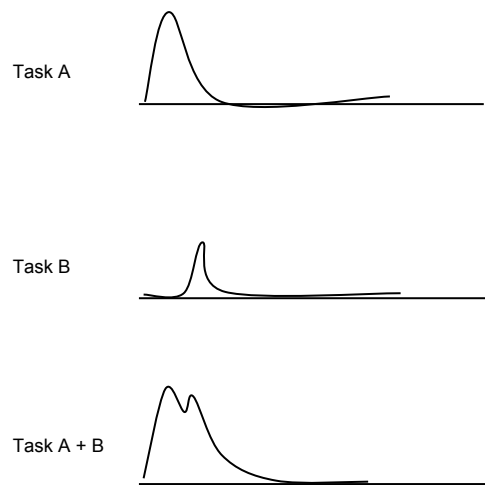


Figure 6.2 – Two tasks of different conditions can generate different BOLD signals (tasks A and B). Task B starts right after task A, but the BOLD signal of task A do not have en ought time to return to baseline before task B starts. When the tasks are too close in time, the resulting BOLD signal of the two tasks sum (Task A + B).

A common approach for estimating one value for a single task is using beta series regression, which is a type of general linear model [RGD04]. Beta series regression calculates one beta value for each task. There are a number of ways of calculating betas. We tried 2 approaches described in the work of Mumford et al. [MTAP12], namely, LS-A and LS-S. LS-A is a traditional approach while LS-S is a new approach they developed and reported as providing better classification accuracy when compared with traditional methods. However, in our experiments we obtained almost identical results using both techniques. Thus, we report only results from the traditional LS-A technique.

6.2.3 Contrast between conditions

For the cross subject experiment, instead of using the time series fMRI data with 640 time points, we use contrast images, that is, a 3D matrix. Contrast images show the intensity difference between two conditions. In this context, we have 4 conditions, regular word, irregular word, pseudo word and baseline. For example, we can create a 3D image that shows the intensity difference between children seeing regular words and seeing irregular words. In this image, each voxel contains a single value which is the intensity difference between the two conditions, regular and irregular words. We call this image *regular > irregular*. Notice image *regular > irregular* is different from image *irregular > regular*. We can create additional contrast images showing the difference between when children are performing any task (summing up the task conditions regular, irregular and pseudo word) and when they are resting: *all > baseline* and *baseline > all*. Figure 6.3 details

how contrast images are generated, using *all* > *baseline* and *baseline* > *all* images as example. This set up results in the following contrast images, which are the main input data for the classifier: *all* > *baseline*, *baseline* > *all*, *regular* > *irregular*, *irregular* > *regular*, *regular* > *pseudo*, *pseudo* > *regular*, *irregular* > *pseudo* and *pseudo* > *irregular*¹.

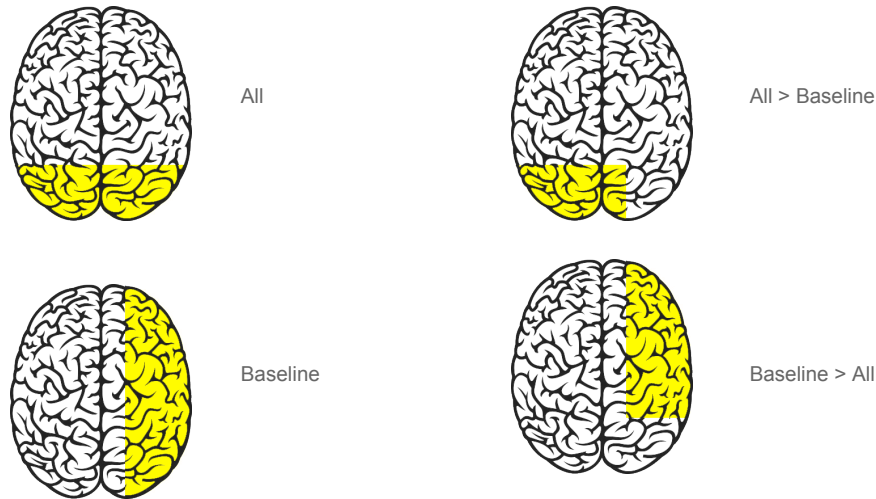


Figure 6.3 – Schematic example of contrast images *all* > *baseline* and *baseline* > *all* generation. First, we calculate the average activation in conditions *all* and in condition *baseline*. In this example, in the left figures, there is more activation in the back of the brain in *all* and more activation in the right hemisphere in *baseline*. Second, we subtract one condition from another, generating the contrast images in the right. For generating *all* > *baseline* image, we subtract *all* from *baseline*. Conversely, for generating *baseline* > *all* image, we subtract *baseline* from *all*.

6.3 Single Subject Experiments

We performed an initial set of experiments classifying tasks within a single subject in order to ensure it is possible to use the feature selection methods in the data. In these experiments, we used feature selection methods from Chapter 4 combined with two different example generation techniques: averaging 4 seconds of the activation images when children are performing a task and performing a beta series regression in the data. As the cross subject experiment is the final deliverable of this work, in this set of experiments we tested only the feature selection methods we can use in both cross subject and single subject experiments.

For this purpose, we used a 3 class Linear Support Vector Machine provided by pyMVPA [HHS⁺09], which is a toolbox in python for classification on fMRI data. The 3 classes are regular, irregular and pseudo word. Children read 20 words of each class in the scanning session, consequently, we have 20 examples of each class, summing 60 examples. We used leave one out cross validation, training with 57 examples and letting 1 example of each class out.

¹The contrast images were kindly provided by Luiz Fernando Dresch

The results of these experiments are summarized in Table 6.1. First, for the Parcellation feature selection, described in Section 4.3, we used cc200 mask and AAL mask, which divided the brain in 190 and 90 features respectively. Second, the whole brain feature selection, described in Section 4.5, extracts all 33697 voxels from the brain and use them as features in the classification. Third, the ROI feature selection, described in Section 4.2, extracts all voxels from each one of the 90 regions of interest defined in AAL mask. The mean classification accuracy of the 10 participants for each ROI is shown in Table A.1 and the average classification accuracy of all ROIs is described in Table 6.1. Finally, the ANOVA feature selection, described in Section 4.6, chooses the 5% most relevant voxels from the brain as features.

	Parcellations – cc200	Parcellations – AAL	Whole Brain	ROI	ANOVA
mean 4 sec	34%	34%	36%	34%	36%
betas	42%	40%	42%	40%	42%

Table 6.1 – Average classification accuracy using data of 10 subjects. The two methods for generating examples are averaging 4 seconds of task and generating betas. The feature selection methods are Parcellations using cc200 and AAL mask (Section 4.3), Whole brain (Section 4.5), ROI (Section 4.2) and ANOVA (Section 4.6).

6.4 Cross Subject Experiments

We performed the final set of experiments classifying different tasks conditions between patients. As in the single subject experiments, we use feature selection methods from Chapter 4 and generated examples from contrast images.

Data from each one of the 10 patients generates 8 contrast images. We created 4 binary classifiers that were trained to distinguish between each pair of different conditions: *All x Baseline*, *Regular x Irregular*, *Regular x Pseudoword* and *Irregular x Pseudoword*. For this purpose, we used contrast images such as (*condition1* > *condition2* and *condition2* > *condition1*), as described in Table 6.2. For training the classifiers, we use 2 contrast images for each patient, summing up 10 contrast images from one class and 10 contrast images from the other class, totalizing 20 images that we transform later in examples.

The classification was performed using leave one patient out cross validation. We trained the classifier with data from 9 subjects and tested it with the data of the remaining subject. We used a Linear Support Vector Machine provided by pyMVPA [HHS⁺09].

The results of these experiments are summarized in Table 6.3. First, for the Parcellation feature selection, described in Section 4.3, we used cc200 mask and AAL mask, which divides the brain in 190 and 90 features respectively. The 10 most important areas for the classification with the best accuracy (cc200) are shown in Table 6.5, these areas are highlighted in Figure 6.4. Second, the whole brain feature selection, described in Section 4.5, extract all the 33697 voxels from the brain and use them as features in the classification. The locations of the most important voxels for the

classifier	class 1	class 2
All x Baseline	all >baseline	baseline >all
Regular x Irregular	regular >irregular	irregular >regular
Regular x pseudoword	regular >pseudo	pseudo >regular
Irregular x Pseudoword	irregular >pseudo	pseudo >irregular

Table 6.2 – Name of classifiers used in the Cross Subject experiments and the contrast files each classifier uses as examples.

	Parcellations – cc200	Parcellations – AAL	Whole Brain	ROI	ANOVA
All x Baseline	100%	100%	100%	80%	100%
Regular x Irregular	90%	50%	100%	61%	90%
Regular x Pseudo	70%	50%	90%	60%	60%
Irregular x Pseudo	90%	80%	80%	90%	90%

Table 6.3 – Average classification accuracy using leave one patient out cross validation with data of 10 subjects. The feature selection methods are Parcellations using cc200 and AAL mask (Section 4.3), Whole brain (Section 4.5), ROI (Section 4.2) and ANOVA (Section 4.6).

classification are listed in Table 6.5, these areas are highlighted in Figure 6.5. Third, the ROI feature selection, described in Section 4.2, extracted all voxels from each 90 region of interest defined in AAL mask. The classification accuracy for each ROI is described in the appendix in Table A.2 and the average classification accuracy of all ROIs is described in Table 6.3. The average classification accuracy of all ROIs using the 4 classifiers is visually described in Figure 6.6. The accuracy of each classifier is shown separately in Figures 6.6 , 6.7, 6.8, 6.9, 6.10. Finally, the ANOVA feature selection, described in Section 4.6, choose the 5% most relevant features from the brain as features.

6.5 Discussion

The single subject experiments did not show promising results in all example generation and feature selection method combinations, for example, the generation method of averaging 4 seconds of task yield poor results independently of the feature selection method used. Indeed, the accuracy for a 3-class classifier is so poor as to be similar to random guessing (33%). The beta generation method yield somewhat better results, which are slightly better than chance than the former technique. However, we could not reach results we consider acceptable with these example generation methods using any feature selection. As we use data from a fast event related paradigm, in which the stimulus duration is 7 seconds and the interval between two stimulus is from 1 to 3 seconds, the BOLD signal does not have time to decay completely after a stimulus is presented, making the BOLD signal of near stimulus overlap. We speculate that the beta generation method provides better results because it obviates the need to separate overlapped signals, making the generated example cleaner. It is known that beta series regression is more suitable for event related data than averaging some data time points [MTAP12].

From the single subject experiments we discovered that the example generation methods create noisy examples. This is because when creating an example, we use the BOLD signal generated when children are reading one word, as we have 60 words, we create 60 examples. However, averaging the BOLD signal of several tasks makes the examples cleaner [BSM⁺12]. For example, we could have averaged two tasks for creating one example, and use 30 cleaner examples for classification. Thus, for the cross subject experiments we use several trials in order to generate cleaner results: each patient generates 1 example of each class using all trials of each class when creating the contrast images. In the cross subject experiment we use 4 classifiers for comparing all different conditions, including the *All* condition summing up all times children are performing a task and *Baseline*, when they are resting. Using the contrast images, we obtained much better accuracy in the cross subject experiments. This result indicates that if the examples are fine, performing feature selection in the data can help in the classification

When analyzing only the classifier results regardless the feature selection used, we supposed the *All x Baseline* classifier would provide the best accuracy. This is the easiest classification in the experiment, used just as sanity check, as the difference between brain patterns when people are resting or performing some task is huge. In the same way, we supposed the *Irregular x Pseudoword* classifier would return the worst result, as those type of words are more difficult to children with dyslexia read. The results in Table 6.3 show a higher accuracy in the *All x Baseline* classification, and surprisingly, a lower accuracy in the *Regular x Pseudoword* classification. The *All x Baseline* result is the highest and the *Regular x Pseudoword* is the lowest in both single and cross subject experiments, showing consistency between the experiment results.

When comparing the feature selection results we can draw a few conclusions about them. First, in the parcellations methods, the results are better with the cc200 mask than with the AAL mask. We speculate that it is because the cc200 mask has double the number of regions of the AAL mask, and because the brain division of cc200 was specifically designed to be used for fMRI data classification. Second, the whole brain feature selection shows the best accuracy of all feature selection methods. This may be due to the broad and sparse reading network configuration. It is known that the reading network in children is not completely formed, and they need to recruit more brain areas than adults for reading task. Further, children with dyslexia recruit even more brain regions for reading tasks than age matched controls. For example, the pseudo word task involves the occipital region for visualizing and decoding a word; the inferior frontal area (Broca's area) for accessing the meaning of the word and deciding if it is a real word or not; the superior temporal and parietal regions to access the pronunciation and articulation of the word; the motor planning area for deciding if one should press the right button if the word is real and the left otherwise; and finally, the motor area for actually pressing a button [Deh09]. Thus, we believe that as several sparse regions in the brain are used for reading, having more voxels from all across the brain provides a better classification accuracy. Third, the ROI feature selection seems not to provide very good results. However, as this result is the average accuracy of all brain ROIs, we need to analyze the results in more detail. For example, although all classifiers have at least one region that returns

100% accuracy, no ROI provides 100% for all 4 classifiers. In fact, the only region that returns 100% accuracy in more than one area is the left insula in the *All x Baseline* and *Regular x Irregular* classifiers. Thus, the ROI feature selection gives us an insight about the brain regions involved in each condition in spite of providing poorer results if used as the only feature for all classification tasks. Fourth, the ANOVA feature selection returns an average result, this might be caused because ANOVA selects only 5% of the voxels from the brain for using in classification. Comparing the ANOVA and whole brain results, we can infer that, in this context, more voxels give us a better classification accuracy. This is the only method that performed well in the single subject experiments and worse in the cross subject experiments.

The important brain regions for all feature selection methods are similar, we enumerate the results that stand out in each classifier.

All x Baseline We expect the *All x Baseline* classifier to choose more vision related regions (in the occipital lobe) as those regions are more requested when children are reading any type of word. The *All x Baseline* classifier chose more occipital regions in both hemispheres independently of the feature selection used, although the chosen regions vary with the feature selection technique. For example, cc200 chose 80% occipital regions, while whole brain and ANOVA chose equally distributed occipital, frontal e superior parietal regions in both hemispheres. All feature selection methods chose left precentral (motor) and inferior frontal (Broca) regions.

Regular x Irregular The *Regular x Irregular* classifier chose more superior parietal and medial frontal regions in both hemispheres. All feature selections chose left or right precentral and inferior frontal regions. The whole feature selection, which had a better accuracy than cc200 and ANOVA, chose more occipital regions in both hemispheres than the other methods.

Regular x Pseudo The *Regular x Pseudoword* classifier, which had the worse accuracy in all feature selection method, chose distinct regions from the previous classifiers. The whole brain method chose much more distributed regions, most of them are in the left parietal region. The left and right cingulate regions were chosen, which were not chosen before, along with the right frontal and basal ganglia regions and left and right occipital regions. cc200 chose similar regions than whole brain feature selection: left parietal and cingulate regions and right basal ganglia regions. An impressive region chosen by cc200 is the right supramarginal gyrus, which in the left hemisphere is known as Wernicke area. ANOVA chose large clusters in fewer regions: left and right precentral, occipital and parietal regions. From the classification results, we can infer that the left and right cingulate and frontal regions and the right basal ganglia regions are important for differentiating between regular and pseudo words, as the whole brain and cc200 feature selection achieved a better accuracy (80% and 70%) using those regions than ANOVA (70%). These results support the idea of a reading network distributed all over the brain, as the classifiers which choose more brain regions obtain better accuracy.

Irregular x Pseudo The *Irregular x Pseudoword* classifiers chose less regions as the most important. All feature selection algorithms chose cingulate and basal ganglial regions in both hemispheres. The whole brain feature selection, which achieved the best accuracy, also chose middle frontal areas in both hemispheres.

All x Baseline	Regular x Irregular	Regular x Pseudoword	Irregular x Pseudoword
L Lingual	L Calcarine	R Precentral	R Thalamus
R Lingual	L Precuneus	L Precuneus	R Precentral
L Precentral	R Precuneus	L Calcarine	L Mid. Orbital
L Fusiform	L Sup. Parietal lobule	L Inf. Parietal lobule	R Lingual
R Fusiform	R Mid. frontal	R Postcentral	R Angular
L Sup. Occipital	R Inf. Frontal (triangularis)	L Sup. Frontal	L Mid. Temporal
L Inf. Frontal	L Sup. Parietal lobule	R Supramarginal gyrus	L Ant. Cingulate Cortex
L Mid. Occipital	R Inf. Frontal (opercularis)	L Sup. Medial	L Thalamus
R Inf. Occipital	R Postcentral	R Hippocampus	L Caudate Nucleus
R Mid. Occipital	R Precentral	L Ant. Cingulate Cortex	L Anterior Cingulate Cortex

Table 6.4 – List of the 10 parcellations that contribute the most to classification using cc200 parcellations.

All x Baseline	Regular x Irregular	Regular x Pseudoword	Irregular x Pseudoword
L Inf occipital	L Precuneus	R precuneus	L sup medial
L Calcarine	R Precuneus	L lingual	R sup medial
L Lingual	L sup parietal	R lingual	L ant cingulate
R inf occipital	R sup parietal	Thalamus	R ant cingulate
R Calcarine	L calcarine	L post cingulate	L mid orbital
R Lingual	L mid occipital	R postero cingulate	R mid orbital
L Precentral	R inf Occipital	L precentral	L precuneus
L Postcentral	R Calcarine	L postcentral	L precentral
L Temporal pole	R mid frontal	R superior frontal	L postcentral
L Inf Frontal	R inf frontal	R middle frontal	L thalamus
L Precuneus	L precentral	R middle orbital	R thalamus
R Precuneus	L Inferior frontal	L inf parietal	R sup frontal
L Sup frontal	L sup medial	L sup occipital	R angular
R Superior Frontal	R sup medial		
L Medial Frontal			
R Medial Frontal			

Table 6.5 – Location of the brain regions that contribute the most for classification using voxels from the whole brain. The regions are the 5% most important voxels that belongs to clusters of at least 100 voxels.

All × Baseline	Regular × Irregular	Regular × Pseudoword	Irregular × Pseudoword
R Inf occipital	L Sup Parietal	R Precentral	R Inf Frontal
L Inf occipital	R Sup Parietal	R Postcentral	R Precentral
L Cuneus	L Precuneus	L Lingual	L Ant Cingulate
R Cuneus	R Precuneus	R Lingual	R Ant cingulate
L Calcarine	L Lingual	L Precentral	L Postcentral
R Calcarine	R Lingual	L Postcentral	R Precentral
L Lingua	R Inf Frontal	L Inf Parietal	R sup Parietal
R Lingual	L Precentral		R Thalamus
R Precuneus	L Inf Frontal		
L Precuneus	L Sup Medial		
L Precentral	R Sup Medial		
L Sup Temporal			
L Inf Frontal			
L Precentral			
L Postcentral			
L Sup frontal			
R Sup frontal			
R Medial Frontal			

Table 6.6 – Location of the brain regions that contribute the most for classification using ANOVA feature selection. The regions are the 5% most important voxels that belongs to clusters of at least 100 voxels.

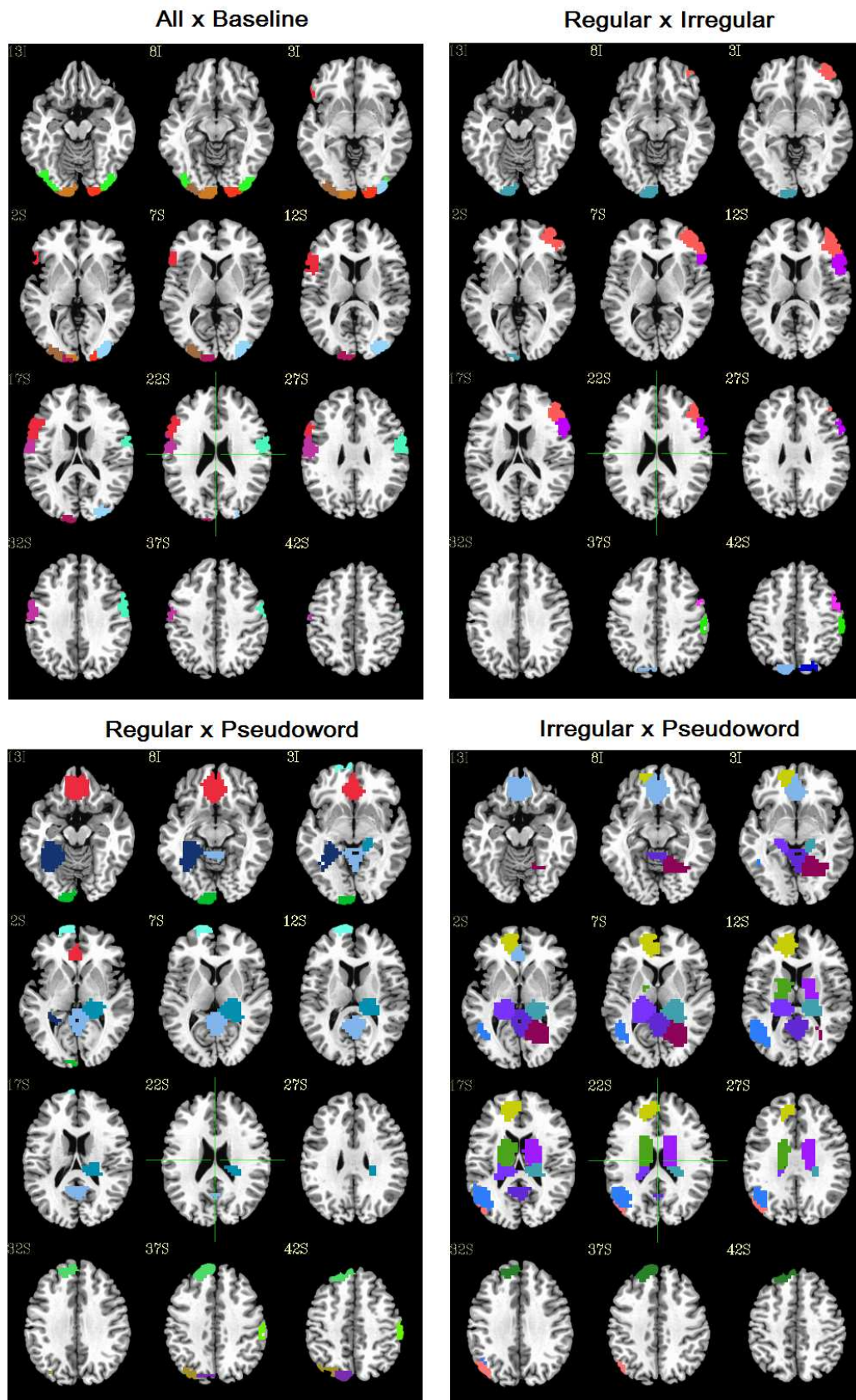


Figure 6.4 – Location of the 10 parcellations that contribute the most to classification from the cc200 parcellations.

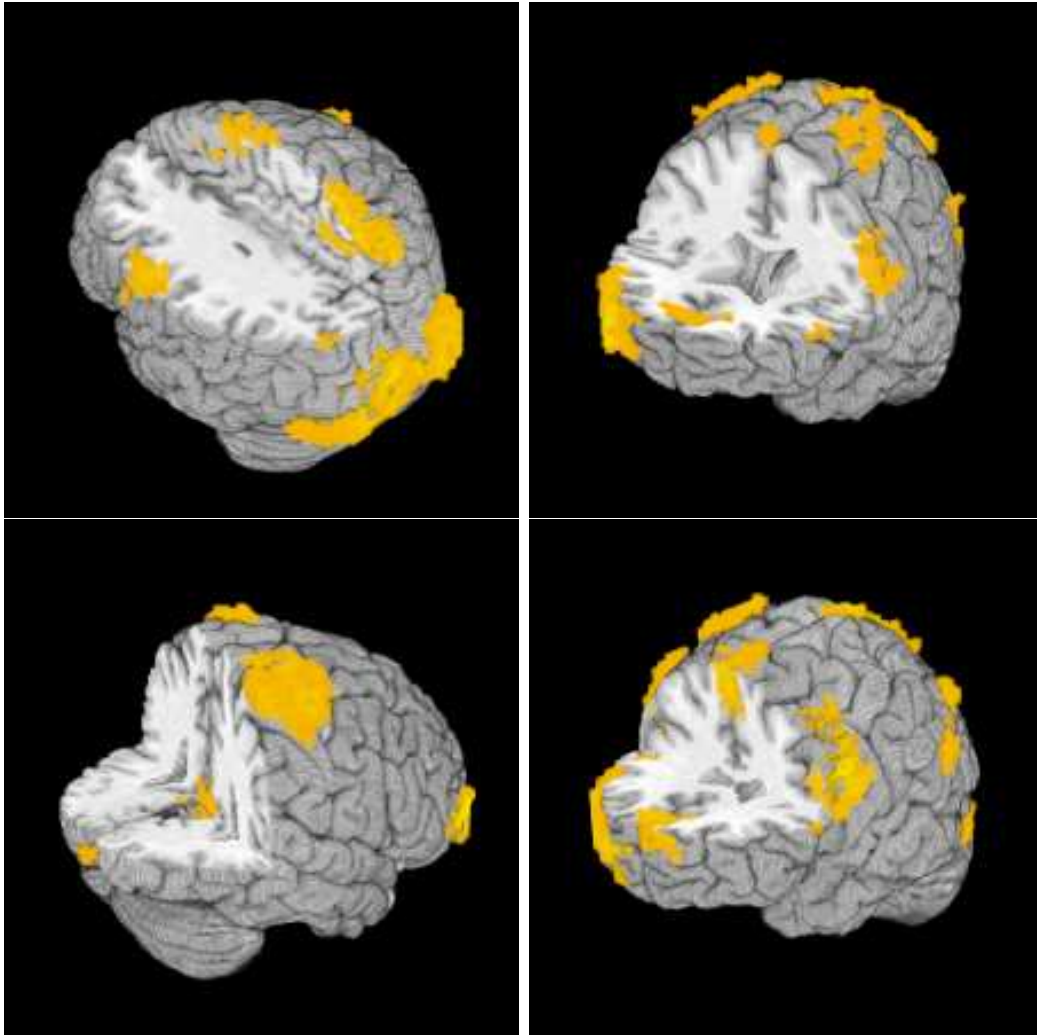


Figure 6.5 – Brain regions that contribute the most for classification using voxels from the whole brain. The regions are the 5% most important voxels that belongs to clusters of at least 100 voxels. Top left: All x Baseline classification; Top right: Regular x Irregular classification; Bottom left: Regular x Pseudo classification; Bottom Right: Irregular x Pseudo classification.

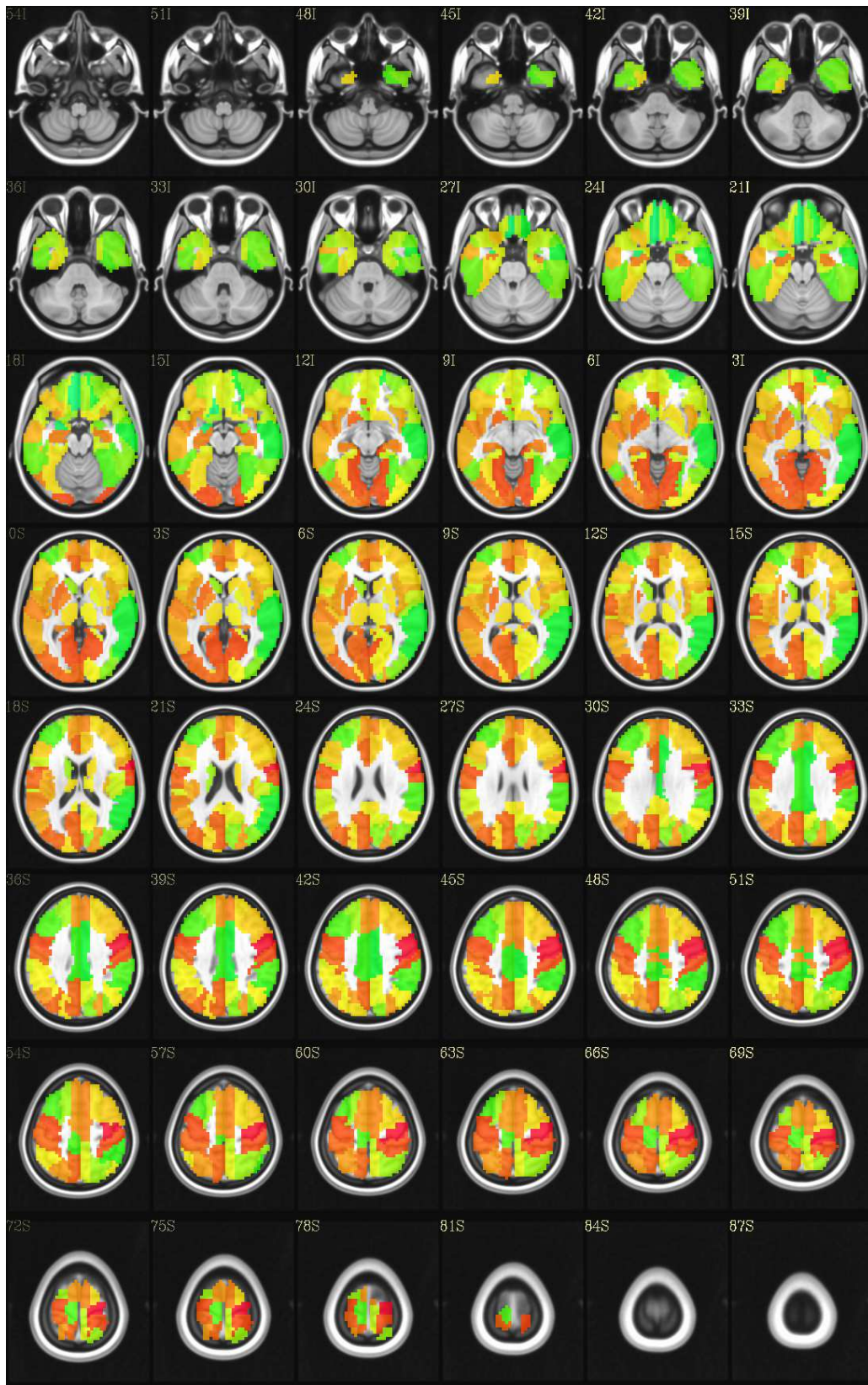


Figure 6.6 – Average classification accuracy from the 4 classifiers defined in Table 6.2. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are left and right precentral and lingual regions and right postcentral region.

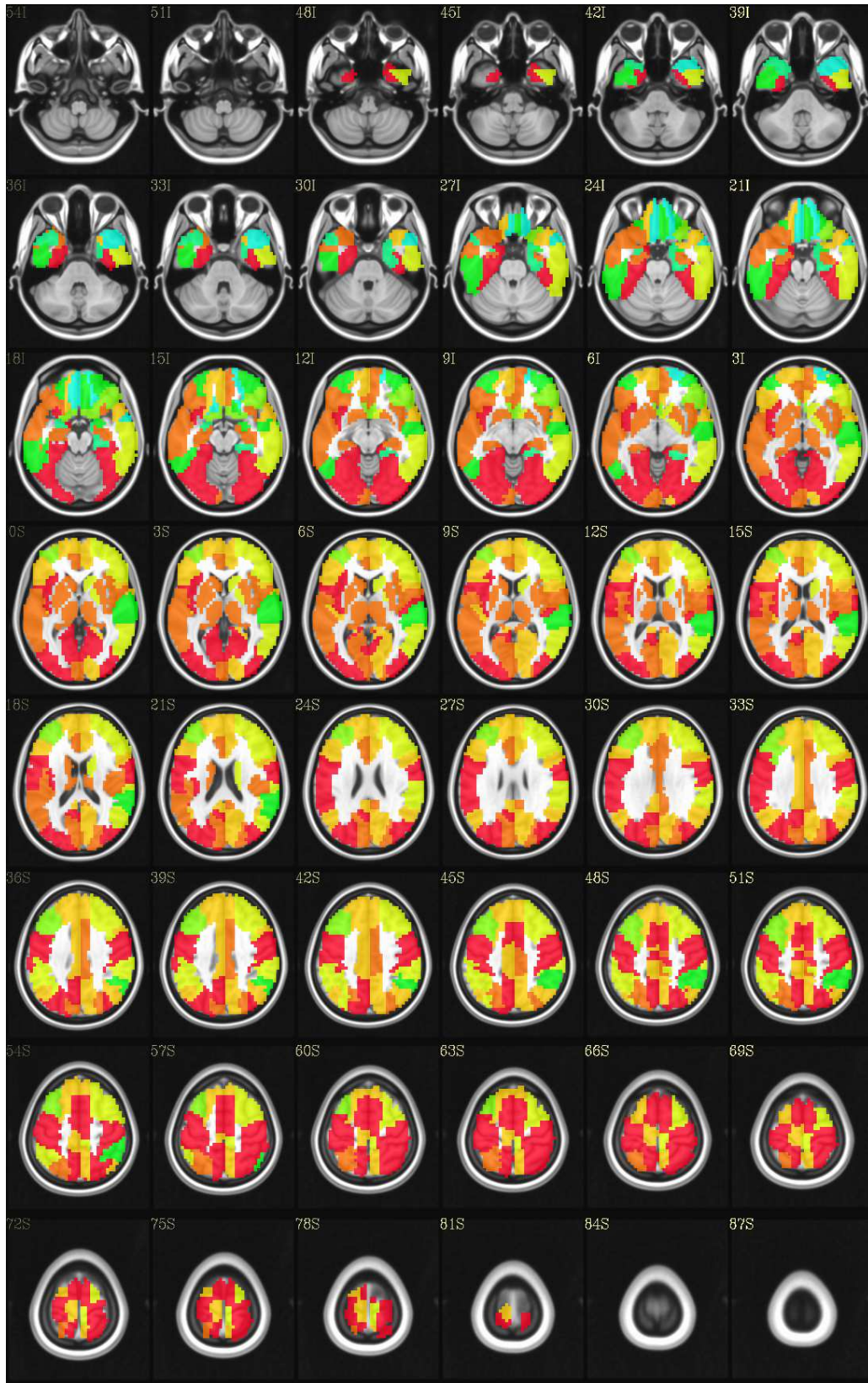


Figure 6.7 – Accuracy from all x baseline classifier. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are left and right occipital and parietal areas, left insula and inferior frontal operculum.

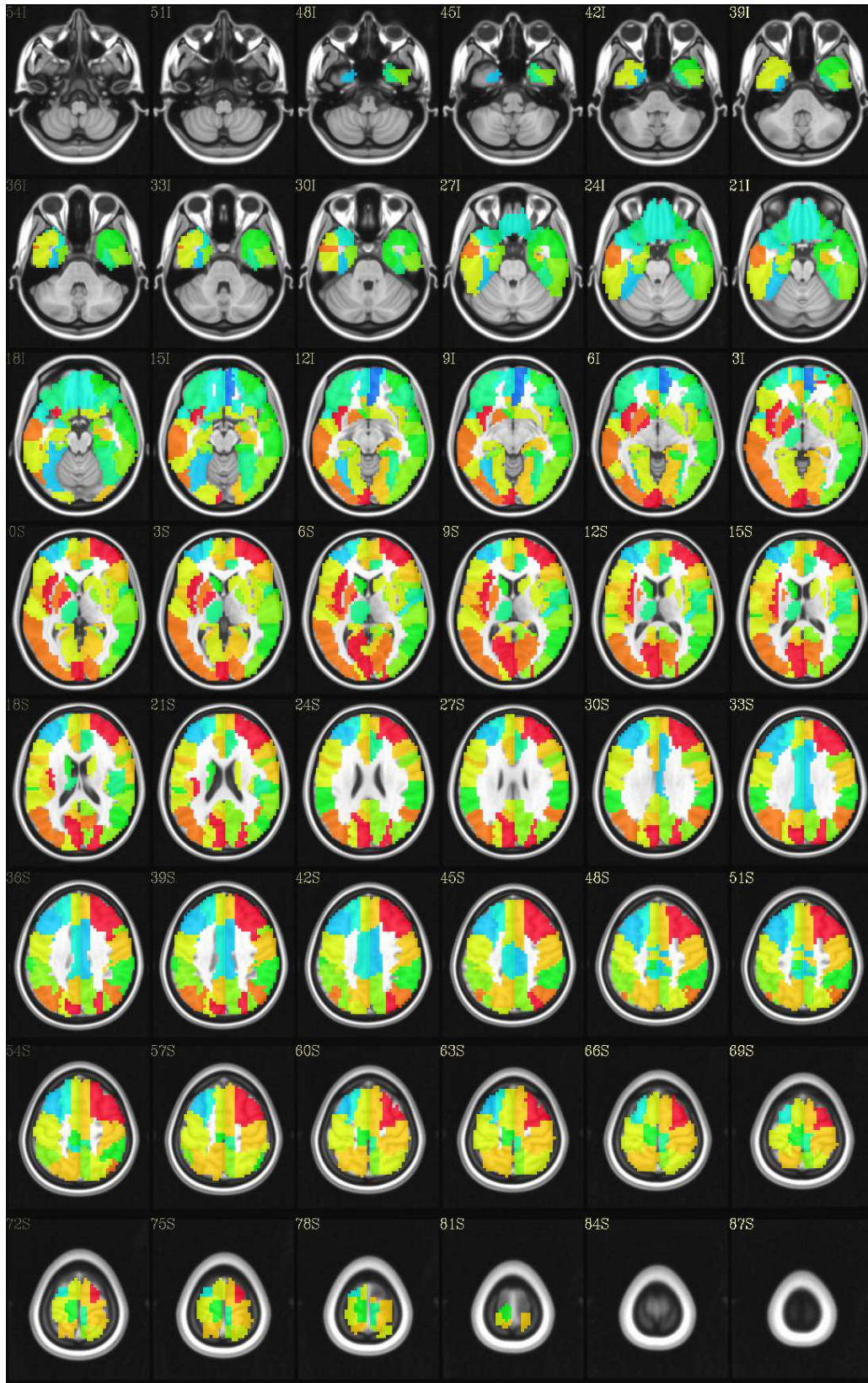


Figure 6.8 – Accuracy from regular x irregular classifier. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are left and right occipital areas, left pallidum and insula and right frontal region.

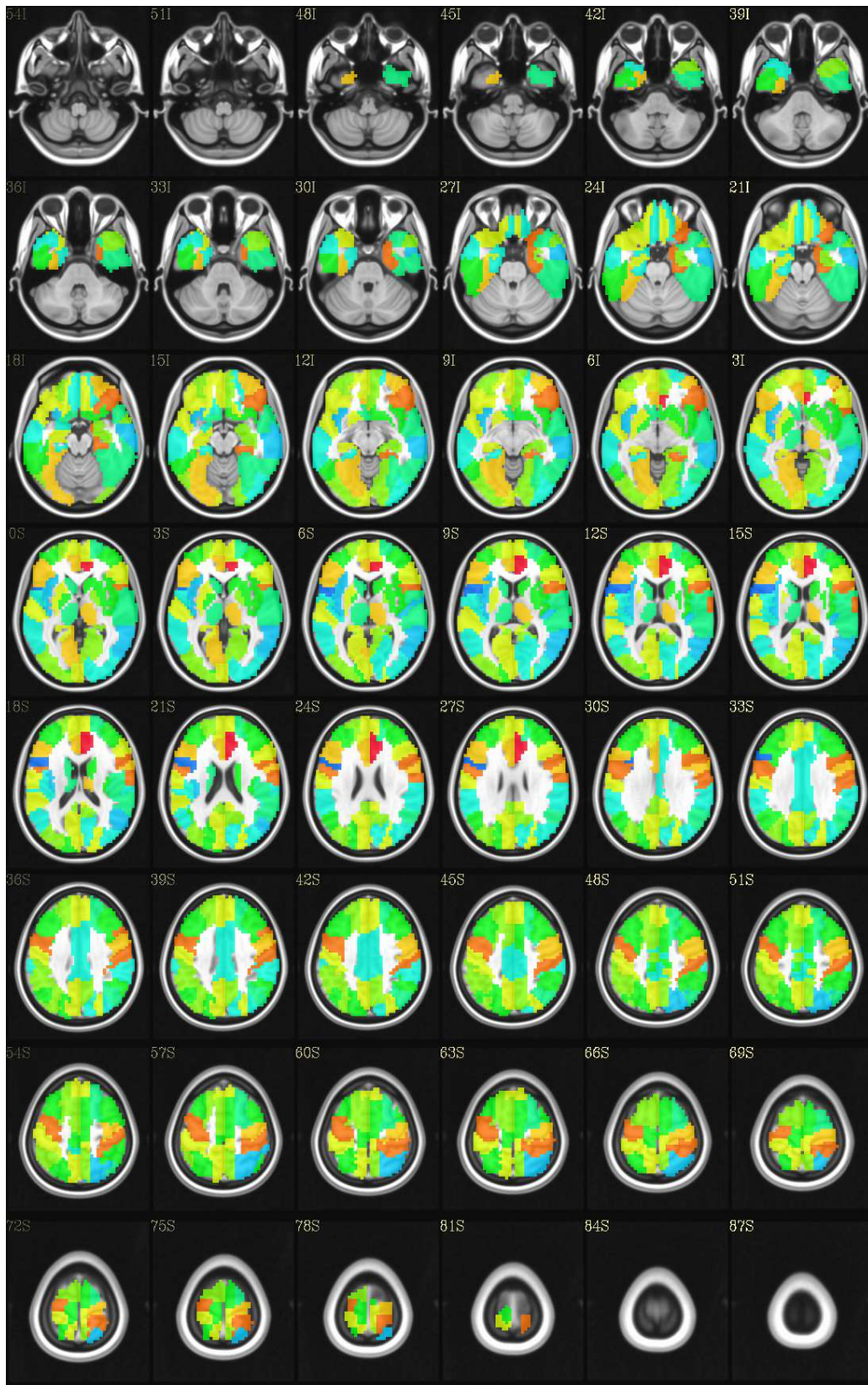


Figure 6.9 – Accuracy from regular x pseudo classifier. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are right inferior frontal, parahippocampal and anterior cingulum and left and right parietal regions.

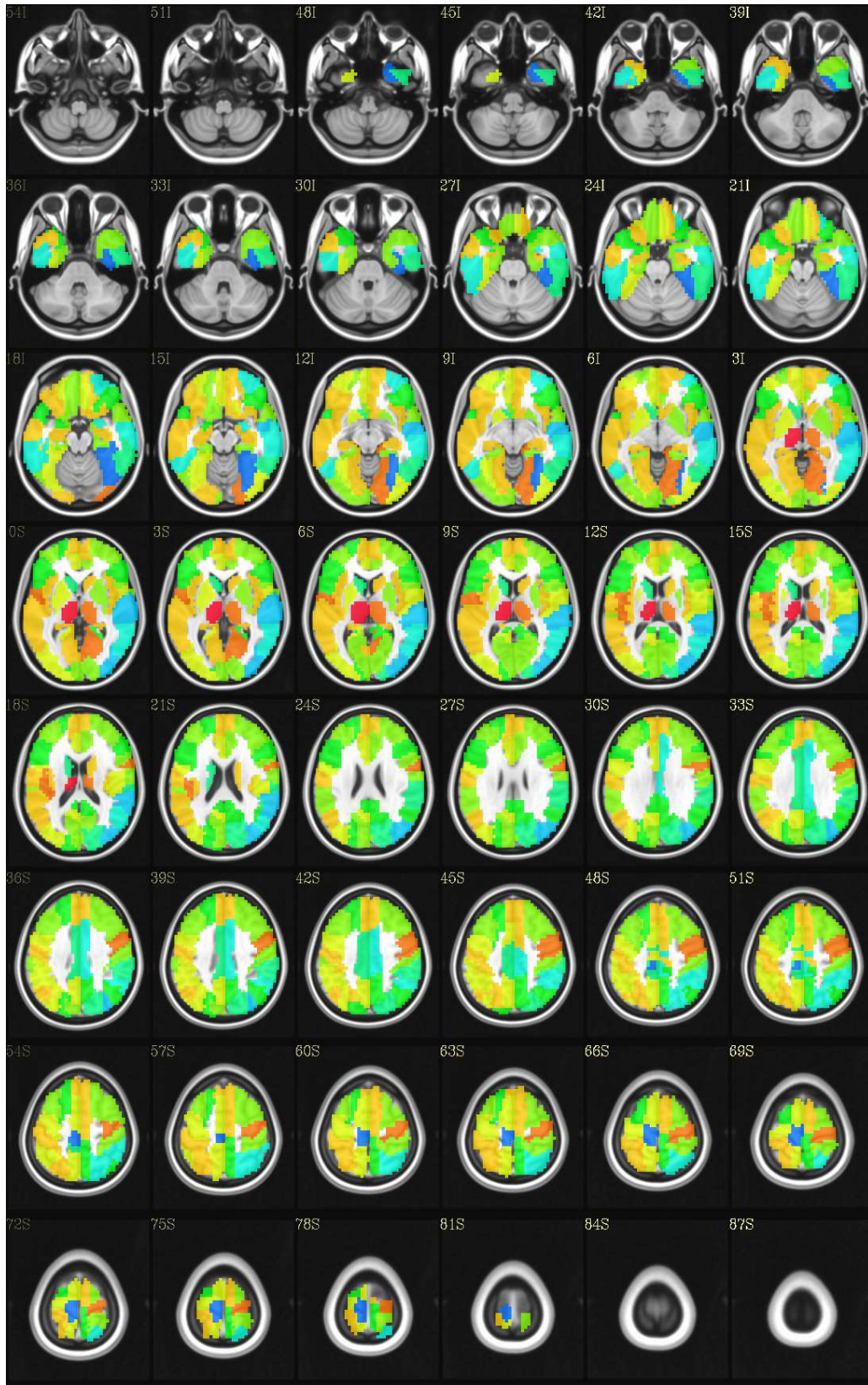


Figure 6.10 – Accuracy from irregular x pseudo classifier. The classification is done in each region of interest defined by AAL mask. Red areas have 100% accuracy and green areas have 0% accuracy. The areas with higher average accuracy are left and right thalamus, right precentral and lingual regions.

7. CONCLUSION

In this work, we have given the initial steps in the development of an fMRI-based technique to diagnose the cognitive states associated with dyslexia. In order to accomplish this, we have researched techniques to process fMRI data so that machine learning techniques can be employed with high accuracy to detect such cognitive states. The techniques we investigated are known as feature selection, which we have systematically used to generate classifiers for word recognition tasks. We empirically tested the techniques using fMRI data from dyslexic children from the ACERTA project. In the experiments we tested which feature selection techniques work better with our data, as well as 3 methods for generating examples. We performed initial experiments using data from one subject at a time in the single subject experiments, and then proceeded to running the cross subject experiment, which is the most significant part of our work. In the cross subject experiments, we tested if our classifier could generalize among subject data and indicate the most important brain regions the classifier use to discriminate the type of word children are reading.

Our results show that it is difficult to use the data of single subjects with classification because the generated examples are too noisy, regardless of the example generation approach. Moreover, the cross subject results show it is easier to work with summarized data of patients by using contrast images, as they generate cleaner examples. We learned that the reading network of our subjects is broad and distributed all over the brain. Consequently, using several voxels from all over the brain, as the whole brain feature selection does, is the best approach for classifying what category of word subjects are reading.

As future work, we plan to use the same data of children with dyslexia along with data from age-matched controls. We aim to use the same classification and feature selection techniques that provided good results in this work to identify differences between the reading network of children with dyslexia and controls. Further, as many neural networks were identified using resting state, including the reading network [BFH⁺99], we want to use the resting state data from children with dyslexia and controls to identify and point the differences between the reading network of both groups.

BIBLIOGRAPHY

- [A⁺13] American Psychiatric Association et al. *The Diagnostic and Statistical Manual of Mental Disorders: DSM 5*. bookpointUS, 2013.
- [ABMSP12] Hiroyuki Akama, Li Na Brian Murphy, Yumiko Shimizu, and Massimo Poesio. Decoding semantics across fmri sessions with different stimulus modalities: a practical mvpa study. *Frontiers in neuroinformatics*, 6, 2012.
- [AGMGVH06] Y Alemán-Gómez, L Melie-García, and P Valdés-Hernandez. Ibaspm: toolbox for automatic parcellation of brain structures. In *12th Annual Meeting of the Organization for Human Brain Mapping*, volume 27, 2006.
- [Alp04] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2004.
- [BFH⁺99] Jeffrey R Binder, Julia A Frost, Thomas A Hammeke, PSF Bellgowan, Stephen M Rao, and Robert W Cox. Conceptual processing during the conscious resting state: a functional mri study. *Journal of cognitive neuroscience*, 11(1):80–93, 1999.
- [BGV92] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.
- [BSM⁺12] Augusto Buchweitz, Svetlana V Shinkareva, Robert A Mason, Tom M Mitchell, and Marcel Adam Just. Identifying bilingual semantic neural representations across languages. *Brain and language*, 120(3):282–289, 2012.
- [BZYHH95] Bharat Biswal, F Zerrin Yetkin, Victor M Haughton, and James S Hyde. Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magnetic resonance in medicine*, 34(4):537–541, 1995.
- [CAH⁺03] Vince D Calhoun, Tülay Adalı, Lars Kai Hansen, Jan Larsen, and James J Pekar. Ica of functional mri data: an overview. 2003.
- [CHHM09] R Cameron Craddock, Paul E Holtzheimer, Xiaoping P Hu, and Helen S Mayberg. Disease state prediction from resting state functional connectivity. *Magnetic resonance in Medicine*, 62(6):1619–1628, 2009.
- [CJH⁺12] R Cameron Craddock, G Andrew James, Paul E Holtzheimer, Xiaoping P Hu, and Helen S Mayberg. A whole brain fmri atlas generated via spatially constrained spectral clustering. *Human brain mapping*, 33(8):1914–1928, 2012.
- [Deh09] Stanislas Dehaene. *Reading in the brain: The new science of how we read*. Penguin, 2009.

- [Dom12] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.
- [EGK09] Joset A Etzel, Valeria Gazzola, and Christian Keysers. An introduction to anatomical roi-based fmri classification analysis. *Brain research*, 1282:114–125, 2009.
- [FHW⁺94] Karl J Friston, Andrew P Holmes, Keith J Worsley, J-P Poline, Chris D Frith, and Richard SJ Frackowiak. Statistical parametric maps in functional imaging: a general linear approach. *Human brain mapping*, 2(4):189–210, 1994.
- [GE03] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [GHG04] Thore Graepel, Ralf Herbrich, and Julian Gold. Learning to fight. In *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, pages 193–200, 2004.
- [Gho04] Imran Ghory. Reinforcement learning in board games. *Department of Computer Science, University of Bristol, Tech. Rep*, 2004.
- [HHS⁺09] Michael Hanke, Yaroslav O Halchenko, Per B Sederberg, Stephen José Hanson, James V Haxby, and Stefan Pollmann. Pymvpa: A python toolbox for multivariate pattern analysis of fmri data. *Neuroinformatics*, 7(1):37–53, 2009.
- [HK00] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
- [HMB⁺11] Fumiko Hoeft, Bruce D McCandliss, Jessica M Black, Alexander Gantman, Nahal Zakerani, Charles Hulme, Heikki Lyytinen, Susan Whitfield-Gabrieli, Gary H Glover, Allan L Reiss, et al. Neural systems predicting long-term outcome in dyslexia. *Proceedings of the National Academy of Sciences*, 108(1):361–366, 2011.
- [HR06] John-Dylan Haynes and Geraint Rees. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, 7(7):523–534, 2006.
- [HSM04] Scott A Huettel, Allen W Song, and Gregory McCarthy. *Functional magnetic resonance imaging*, volume 1. Sinauer Associates Sunderland, 2004.
- [HTF01] Trevor Hastie, Robert Tibshirani, and J Jerome H Friedman. *The elements of statistical learning*, volume 1. Springer New York, 2001.
- [JKM⁺12] Marcel Adam Just, Timothy A Keller, Vicente L Malave, Rajesh K Kana, and Sashank Varma. Autism as a neural systems disorder: a theory of frontal-posterior underconnectivity. *Neuroscience & Biobehavioral Reviews*, 36(4):1292–1313, 2012.

- [Joh01] Roger W Johnson. An introduction to the bootstrap. *Teaching Statistics*, 23(2):49–54, 2001.
- [Ken90] Maurice George Kendall. Rank correlation methods. 1990.
- [LLL⁺06] Haihong Liu, Zhening Liu, Meng Liang, Yihui Hao, Lihua Tan, Fan Kuang, Yanhong Yi, Lin Xu, and Tianzi Jiang. Decreased regional homogeneity in schizophrenia: a resting state functional magnetic resonance imaging study. *Neuroreport*, 17(1):19–22, 2006.
- [LRM⁺11] Quoc V Le, Marc’Aurelio Ranzato, Rajat Monga, Matthieu Devin, Kai Chen, Greg S Corrado, Jeff Dean, and Andrew Y Ng. Building high-level features using large scale unsupervised learning. *arXiv preprint arXiv:1112.6209*, 2011.
- [Maz09] Alessandro A Mazzola. Ressonância magnética: princípios de formação da imagem e aplicações em imagem funcional. *Revista Brasileira de Física Médica*, 3(1):117–29, 2009.
- [Mit97] Tom M Mitchell. Machine learning. *Burr Ridge, IL: McGraw Hill*, 45, 1997.
- [MSC⁺08] Tom M Mitchell, Svetlana V Shinkareva, Andrew Carlson, Kai-Min Chang, Vicente L Malave, Robert A Mason, and Marcel Adam Just. Predicting human brain activity associated with the meanings of nouns. *science*, 320(5880):1191–1195, 2008.
- [MTAP12] Jeanette A Mumford, Benjamin O Turner, F Gregory Ashby, and Russell A Poldrack. Deconvolving bold activation in event-related designs for multivoxel pattern classification analyses. *NeuroImage*, 59(3):2636–2643, 2012.
- [PMB09] Francisco Pereira, Tom Mitchell, and Matthew Botvinick. Machine learning classifiers and fMRI: A tutorial overview. *NeuroImage*, 45(1, Supplement 1):S199–S209, 2009. *Mathematics in Brain Imaging*.
- [PTP⁺01] Russel Poldrack, Elise Temple, Athanassios Protopapas, Srikantan Nagarajan, Paula Tallal, Michael Merzenich, and J Gabrieli. Relations between the neural bases of dynamic auditory processing and phonological processing: evidence from fmri. *Cognitive Neuroscience, Journal of*, 13(5):687–697, 2001.
- [PVG⁺11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [RGD04] Jesse Rissman, Adam Gazzaley, and Mark D’Esposito. Measuring functional connectivity during distinct stages of a cognitive task. *Neuroimage*, 23(2):752–763, 2004.

- [RN10] Stuart Jonathan Russell and Peter Norvig. *Artificial intelligence: a modern approach*, volume 2. Prentice hall, 2010.
- [SC14] Xiaomu Song and Nan-kuei Chen. A unified machine learning method for task-related and resting state fmri data analysis. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 6426–6429. IEEE, 2014.
- [SCK⁺14] S Sikka, B Cheung, R Khanuja, S Ghosh, C Yan, Q Li, J Vogelstein, R Burns, S Colcombe, C Craddock, et al. Towards automated analysis of connectomes: The configurable pipeline for the analysis of connectomes (c-pac). In *Front. Neuroinform. Conference Abstract: 5th INCF Congress of Neuroinformatics*. doi: 10.3389/conf.fninf, volume 117, 2014.
- [SEVA09] John T Serences, Edward F Ester, Edward K Vogel, and Edward Awh. Stimulus-specific delay activity in human primary visual cortex. *Psychological Science*, 20(2):207–214, 2009.
- [Sha08] Sally Shaywitz. *Overcoming dyslexia: A new and complete science-based program for reading problems at any level*. Random House LLC, 2008.
- [SMM⁺08] Svetlana V Shinkareva, Robert A Mason, Vicente L Malave, Wei Wang, Tom M Mitchell, and Marcel Adam Just. Using fmri brain activation to identify cognitive states associated with perception of tools and dwellings. *PLoS One*, 3(1):e1394, 2008.
- [SMvdM⁺08] Enrico Schulz, Urs Maurer, Sanne van der Mark, Kerstin Bucher, Silvia Brem, Ernst Martin, and Daniel Brandeis. Impaired semantic processing during sentence reading in children with dyslexia: Combined fmri and erp evidence. *Neuroimage*, 41(1):153–168, 2008.
- [Tay11] Matthew E Taylor. Teaching reinforcement learning with mario: An argument and case study. In *Second AAAI Symposium on Educational Advances in Artificial Intelligence*, 2011.
- [TSK05] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley, 2005.
- [Vap00] Vladimir Vapnik. *The nature of statistical learning theory*. springer, 2000.
- [WS12] John L Woodard and Michael A Sugarman. Functional magnetic resonance imaging in aging and dementia: Detection of age-related cognitive changes and prediction of cognitive decline. In *Behavioral Neurobiology of Aging*, pages 113–136. Springer, 2012.

- [WSN⁺09] JL Woodard, M Seidenberg, KA Nielson, P Antuono, L Guidotti, S Durgerian, Q Zhang, M Lancaster, N Hantke, A Butts, et al. Semantic memory activation in amnesic mild cognitive impairment. *Brain*, 132(8):2068–2078, 2009.
- [ZJL⁺04] Yufeng Zang, Tianzi Jiang, Yingli Lu, Yong He, and Lixia Tian. Regional homogeneity approach to fmri data analysis. *Neuroimage*, 22(1):394–400, 2004.

APPENDIX A – CLASSIFICATION ACCURACY IN EACH ROI

Table A.1: Average classification accuracy of single subject classification using all voxels of each region of interest. ROIs are defined in AAL mask. Two methods are used to generate examples: betas and averaging 4 seconds of stimulus.

label	betas	mean 4 s
Precentral L	0.43	0.34
Precentral R	0.44	0.37
Frontal Sup L	0.41	0.33
Frontal Sup R	0.39	0.34
Frontal Sup Orb L	0.39	0.34
Frontal Sup Orb R	0.41	0.34
Frontal Mid L	0.39	0.34
Frontal Mid R	0.40	0.34
Frontal Mid Orb L	0.41	0.31
Frontal Mid Orb R	0.42	0.35
Frontal Inf Oper L	0.40	0.37
Frontal Inf Oper R	0.40	0.35
Frontal Inf Tri L	0.44	0.32
Frontal Inf Tri R	0.38	0.33
Frontal Inf Orb L	0.42	0.37
Frontal Inf Orb R	0.39	0.35
Rolandic Oper L	0.39	0.36
Rolandic Oper R	0.43	0.34
Supp Motor Area L	0.43	0.37
Supp Motor Area R	0.40	0.36
Olfactory L	0.37	0.31
Olfactory R	0.39	0.35
Frontal Sup Medial L	0.41	0.36
Frontal Sup Medial R	0.41	0.35
Frontal Mid Orb L	0.41	0.35
Frontal Mid Orb R	0.42	0.35
Rectus L	0.40	0.34
Rectus R	0.40	0.37
Insula L	0.40	0.34
Insula R	0.40	0.35

Continued on next page

Table A.1 – continued from previous page

label	betas	mean 4 s
Cingulum Ant L	0.38	0.36
Cingulum Ant R	0.36	0.35
Cingulum Mid L	0.42	0.33
Cingulum Mid R	0.40	0.38
Cingulum Post L	0.41	0.35
Cingulum Post R	0.41	0.35
Hippocampus L	0.42	0.36
Hippocampus R	0.40	0.34
ParaHippocampal L	0.42	0.37
ParaHippocampal R	0.38	0.37
Amygdala L	0.40	0.34
Amygdala R	0.40	0.35
Calcarine L	0.43	0.34
Calcarine R	0.41	0.34
Cuneus L	0.41	0.34
Cuneus R	0.38	0.35
Lingual L	0.41	0.36
Lingual R	0.39	0.32
Occipital Sup L	0.42	0.32
Occipital Sup R	0.41	0.37
Occipital Mid L	0.40	0.33
Occipital Mid R	0.37	0.33
Occipital Inf L	0.41	0.25
Occipital Inf R	0.39	0.31
Fusiform L	0.42	0.33
Fusiform R	0.40	0.30
Postcentral L	0.42	0.36
Postcentral R	0.44	0.34
Parietal Sup L	0.40	0.32
Parietal Sup R	0.37	0.37
Parietal Inf L	0.39	0.35
Parietal Inf R	0.40	0.37
SupraMarginal L	0.40	0.34
SupraMarginal R	0.42	0.34
Angular L	0.44	0.31
Angular R	0.41	0.34
Precuneus L	0.40	0.34

Continued on next page

Table A.1 – continued from previous page

label	betas	mean 4 s
Precuneus R	0.41	0.35
Paracentral Lobule L	0.40	0.37
Paracentral Lobule R	0.39	0.33
Caudate L	0.41	0.33
Caudate R	0.41	0.33
Putamen L	0.41	0.35
Putamen R	0.42	0.35
Pallidum L	0.38	0.32
Pallidum R	0.39	0.31
Thalamus L	0.41	0.36
Thalamus R	0.39	0.35
Heschl L	0.41	0.33
Heschl R	0.41	0.30
Temporal Sup L	0.43	0.30
Temporal Sup R	0.38	0.34
Temporal Pole Sup L	0.39	0.31
Temporal Pole Sup R	0.40	0.36
Temporal Mid L	0.44	0.33
Temporal Mid R	0.38	0.34
Temporal Pole Mid L	0.40	0.31
Temporal Pole Mid R	0.42	0.32
Temporal Inf L	0.41	0.33
Temporal Inf R	0.42	0.33
mean	0.40	0.34

Table A.2: Classification accuracy of cross subject classification using all voxels of each region of interest. 4 classifiers are used. ROIs are defined in AAL mask.

label	All x Bas.	Reg. x Irr.	Irr. x Pse.	Reg. x Pse.	Mean
label	All x Baseline	Regular x Irregular	Irregular x Pseudo	Regular x Pseudo	Mean
Precentral R	1	0.8	0.9	0.8	0.875
Lingual R	1	0.8	0.9	0.6	0.825
Precentral L	1	0.7	0.7	0.9	0.825
Lingual L	1	0.7	0.8	0.8	0.825

Continued on next page

Table A.2 – continued from previous page

label	All x Bas.	Reg. x Irr.	Irr. x Pse.	Reg. x Pse.	Mean
Postcentral R	1	0.8	0.6	0.9	0.825
Cingulum Ant L	0.9	0.8	0.7	0.8	0.8
Calcarine L	0.9	1	0.6	0.7	0.8
Cuneus L	0.9	1	0.6	0.7	0.8
Postcentral L	1	0.7	0.8	0.7	0.8
Precuneus L	1	0.8	0.7	0.7	0.8
Putamen L	0.9	0.9	0.7	0.7	0.8
Occipital Mid L	1	0.9	0.7	0.6	0.8
Frontal Sup Medial R	0.8	0.8	0.8	0.7	0.775
Hippocampus L	0.9	0.7	0.8	0.7	0.775
Temporal Sup L	0.9	0.7	0.8	0.7	0.775
Supp Motor Area R	1	0.8	0.8	0.5	0.775
Hippocampus R	0.9	0.8	0.8	0.6	0.775
Occipital Inf L	1	0.9	0.6	0.6	0.775
Pallidum L	0.9	1	0.7	0.5	0.775
Supp Motor Area L	1	0.7	0.7	0.6	0.75
Frontal Sup Medial L	0.8	0.7	0.8	0.7	0.75
Insula L	1	1	0.8	0.2	0.75
Occipital Sup R	0.9	1	0.4	0.7	0.75
Parietal Sup L	0.9	0.8	0.8	0.5	0.75
Angular L	0.8	0.9	0.8	0.5	0.75
Insula R	0.9	0.7	0.8	0.5	0.725
Frontal Inf Oper R	0.8	0.6	0.6	0.9	0.725
Rolandic Oper L	0.9	0.8	0.9	0.3	0.725
Temporal Mid L	0.9	0.9	0.8	0.3	0.725
Cingulum Ant R	0.8	0.4	0.6	1	0.7
Frontal Mid R	0.7	1	0.6	0.5	0.7
Frontal Inf Tri L	0.8	0.7	0.5	0.8	0.7
Frontal Inf Orb L	0.9	0.4	0.8	0.7	0.7
Heschl L	0.8	0.9	0.8	0.3	0.7
Frontal Inf Tri R	0.7	0.8	0.5	0.7	0.675
Cingulum Post R	0.9	0.6	0.5	0.7	0.675
Occipital Sup L	1	0.7	0.5	0.5	0.675
Fusiform L	1	0.2	0.7	0.8	0.675
Caudate R	0.7	0.7	0.8	0.5	0.675
Putamen R	0.9	0.7	0.6	0.5	0.675
Frontal Sup R	0.7	1	0.6	0.4	0.675

Continued on next page