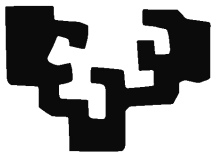


Contributions of Lattice Computing to Medical Image Procesing

By

Darya Chyzhyk

Submitted to the department of Computer Science and Artificial Intelligence in partial fulfillment of the requirements for the degree of Doctor of Philosophy

PhD Advisor:

Prof. Dr. Manuel Graña Romay At The University of the Basque Country

Universidad del País Vasco
Euskal Herriko Unibertsitatea
Donostia - San Sebastian

2013

Universidad del País Vasco Euskal Herriko Unibertsitatea

AUTORIZACION DEL/LA DIRECTOR/A DE TESIS

PARA SU PRESENTACION

Dr/a. _____con N.I.F._____

como Director/a de la Tesis Doctoral: _____

_____

_____

realizada en el Departamento _____

_____

por el Doctorando Don/ña. _____ ,

autorizo la presentación de la citada Tesis Doctoral, dado que reúne las condiciones necesarias para su defensa.

En _____a _____de _____de _____

EL/LA DIRECTOR/A DE LA TESIS

Fdo.: _____

Universidad del País Vasco Euskal Herriko Unibertsitatea

CONFORMIDAD DEL DEPARTAMENTO

El Consejo del Departamento de _____

_____

en reunión celebrada el día ____ de _____ de _____ ha acordado dar la conformidad a la admisión a trámite de presentación de la Tesis Doctoral titulada: _____

_____

_____

dirigida por el/la Dr/a. _____

_____

y presentada por Don/ña. _____

ante este Departamento.

En _____a _____de _____de _____

Vº Bº DIRECTOR/A DEL DEPARTAMENTO SECRETARIO/A DEL DEPARTAMENTO

Fdo.: _____ Fdo.: _____

Universidad del País Vasco Euskal Herriko Unibertsitatea

ACTA DE GRADO DE DOCTOR

ACTA DE DEFENSA DE TESIS DOCTORAL

DOCTORANDO DON/ÑA. _____

TITULO DE LA TESIS: _____

_____

_____

El Tribunal designado por la Subcomisión de Doctorado de la UPV/EHU para calificar la Tesis Doctoral arriba indicada y reunido en el día de la fecha, una vez efectuada la defensa por el doctorando y contestadas las objeciones y/o sugerencias que se le han formulado, ha otorgado por_____la calificación de:

unanimidad ó mayoría

Idioma/s defensa: _____

En _____a _____de _____de _____

EL/LA PRESIDENTE/A, EL/LA SECRETARIO/A,

Fdo.: Fdo.:

Dr/a: _____ Dr/a: _____

VOCAL 1º, VOCAL 2º, VOCAL 3º,

Fdo.: Fdo.: Fdo.:

Dr/a: _____Dr/a:_____Dr/a:_____

EL/LA DOCTORANDO/A,

Fdo.: _____

Contributions of Lattice Computing to Medical Image Procesing

by
Darya Chyzhyk

Submitted to the Department of Computer Science and Artificial Intelligence, in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Abstract

This Thesis is developed along two main axis: The exploration of new computational solutions based on the novel paradigm of Lattice Computing. The application to medical image data in order to obtain new image processing methods, or computed aided diagnosis systems based on image biomarkers. he proposal of Lattice Computing encompasses all computational constructions involving the use of Lattice Theory results and/or operators. In this Thesis, this ambitious scope is reduced to three fields of development of algorithms: Lattice Associative Memories, Dendritic Computing, and Multivariate Mathematical Morphology. Specifically, Lattice Auto-associative Memories play a role in the development of a Lattice Independent Component Analysis (LICA) proposed as a lattice based alternative to the well known Independent Component Analysis (ICA) approach, and in the definition of a reduced ordering in an approach to define a Multivariate Mathematical Morphology. These issues have been tackled in this Thesis from an application point of view: the diverse tools are applied to several kinds of medical image data. From the medical image processing point of view, the Thesis works on features of anatomical MRI for computer aided diagnosis of Alzheimer's disease, on resting state fMRI of Schizophrenia patients, and on CTA data for Abdominal Aortic Aneurysm. **Keywords**: *Medical Image Processing, Lattice Computing, Dendritic Computing, Lattice Independent Component Analysis, Active Learning, Ensemble classifiers*.

Agradecimientos

Me gustaría dar las gracias a todas las personas que, de una forma u otra, han participado de alguna forma en la consecución de esta Tesis. Me gustaría agradecer de manera especial a mi director de Tesis, el profesor Manuel Graña por su asesoramiento, experiencia .....

....

.....

.....

Darya Chyzhyk

Acknowledgements

I want to thank everybody involved in any way with my Thesis and my life. My special thanks to my advisor, Prof. Manuel Graña. Without him this Thesis . .....

...

,,,,

,,,,,

Darya Chyzhyk

To my parents

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This Chapter provides a general introduction to the Thesis, giving a motivation in Section 1.1, an overview of the Thesis contents and contributions in Section 1.2, the publications obtained along the Thesis works in Section 1.3, and finally, the structure of the Thesis in Section 1.4.

1.1 Motivation

This Thesis is developed along two main axis:

- The exploration of new computational solutions based on the novel paradigm of Lattice Computing [**?**] .
- The application to medical image data in order to obtain new image processing methods, or computed aided diagnosis systems based on image biomarkers.

1.1.1 Lattice Computing

The proposal of Lattice Computing encompasses all computational constructions involving the use of Lattice Theory results and/or operators. In this Thesis, this ambitious scope is reduced to three fields of development of algorithms: Lattice Associative Memories, Dendritic Computing, and Multivariate Mathematical Morphology. Specifically, Lattice Auto-associative Memories play a role in the development of a Lattice Independent Component Analysis (LICA) proposed as a lattice

based alternative to the well known Independent Component Analysis (ICA) approach, and in the definition of a reduced ordering in an approach to define a Multivariate Mathematical Morphology. These issues have been tackled in this Thesis from an application point of view: the diverse tools are applied to several kinds of medical image data.

Lattice based Neural Networks (LNNs) - although not yet recognized as mainstream in machine learning - have become an integral part of artificial neural network theory [**?**, **?**]. One reason for this is their simplicity and fast learning methods and another is due to their successful applicability in several disciplines [**?**, **?**, **?**, **?**, **?**, **?**]. Dendritic Computing refers to the construction using Lattice operators of Single Layer Neural Networks. The work on Dendritic Computing deals with new algorithms and improvements over already proposed algorithms found in the literature, also demonstrated over medical image data. The departure point of our work has been the realization that reported algorithms have excelent function approximation performance, they are able to approach exactly any functional relationship between the data, but not so good generalization performance, that is, they degrade on testing data. Therefore, we have combine Dendritic Computing with other computational approaches, such as LICA, including ensamble realizations.

1.1.2 Medical Image Processing

Alzheimer's Disease computer aided diagnosis Many classification experiments in this Thesis are carried out on an specific dataset that was built to explore the realization of computer aided diagnosis systems for Alzheimer's Disease on the basis of features extracted from anatomical MRI brain volumes. The feature extraction process and the details of the population are given in Appendix A. The results obtained have limited value as demonstrators of the feasibility of such computer aided diagnosis systems, however the dataset is valuable as a benchmark dataset because we have already applied a large variety of algorithms over it. Besides, we have also used the same collection of MRI volumes to test alternative ways to perform Voxel Based Morphometry (VBM).

fMRI data We have used synthetic fMRI data for the exploration of the LICA approach compared with competing ICA approaches. Besides, in a more realistic setting, we have make studies on resting state fMRI data from a population of healthy controls and Schizophrenia patients with and without auditory hallucina-

tions. We have found some discriminant information following the LICA approach. Also, the Multivariate Mathematica Morphology approach has been useful to find discriminant connectivity information.

CTA image segmentation The specific problem is the development of classifiers to perform Computed Tomography Angiography (CTA) image segmentation for Abdominal Aortic Aneurysm (AAA) patients. Te classifiers are tailored to each volume by an Active Learning process, which uses the Boostratped Dendritic Computing as the base classifier. Application of the approach requires the definition of appropriate image features, as well as a classification uncertainty measure that guides the addition of samples to the training set.

The specific contributions of the approach proposed in this Thesis relative to the state of the art of AAA thrombus segmentation algorithms are: (1) the need for human intervention in the selection of samples and labeling is reduced to a minimum by Active Learning, (2) the BDC allows quick learning and adaptation as well as the definition of an appropriate uncertainty measure, (3) there is no requirement of a priori information or geometric models, (4) feature extraction does not require sophisticated data processing, (5) the adaptation of the classifier to new data does not require skillful data processing, only picking the most uncertain voxels over a data visualization.

1.2 Overview of the Thesis Contributions

Dendritic computing Dendritic Computing is based on the concept that dendrites are the basic building blocks for a wide range of nervous systems. Dendritic Computing has been proved to achieve perfect approximation of any data distribution, which guarantees perfect accuracy training. However, we found great performance degradation when tested on conventional k-fold cross-validation schemes. In order to improve generalization we have followed various paths:

- Modifying the hyperbox by a reduction factor, which relaxes perfect approximation to obtain some improvement in the testing phase.
- Performing appropriate combination with data transformations, specifically with the LICA approach and a kernelization of the data.
- Composing a collection of weak Dendritic Classifiers into and ensemble by majority voting, with we call Boostraped Dendritic Computing.

As part of the collaboration with professor Ritter, we have also explored the feasibility of building robust associative memories based on Dendritic Computing layers, testing them on a collection of heavily corrupted images.

LICA The Lattice Independent Component Analysis (LICA) [**?**, **?**] consists of two steps. Firts it selects Strong Lattice Independent (SLI) vectors from the input dataset using an incremental algorithm, such as the Incremental Endmember Induction Algorithm (IEIA) [**?**]. Second, because of the conjectured equivalence between SLI and Affine Independence [**?**], it performs the linear unmixing of the input dataset based on these endmembers[1]. Therefore, the approach is a mixture of linear and nonlinear methods. We assume that the data is generated as a convex combination of a set of endmembers which are the vertices of a convex polytope covering some region of the input data. This assumption is similar to the linear mixture assumed by the ICA approach, however LICA does not impose any probabilistic assumption on the data. The endmembers discovered by the IEIA are equivalent to the General Linear Model (GLM) design matrix columns, and the unmixing process is identical to the conventional least squares estimator so LICA is a kind of unsupervised GLM whose regressor functions are mined from the input dataset. If we try to stablish correspondences to the Independent Component Analysis (ICA), the LICA endmembers correspond to the unknown sources and the mixing matrix is the one given by the abundance coefficients computed by least squares estimation. LICA is unsupervised, as ICA, and it does not impose any a priori assumption on the data, thus avoiding "double dipping" effects biasing the results. We have applied LICA to analysis of synthetic fMRI data [**?**, **?**], to Voxel Based Morphometry of structural MRI [**?**], and to resting state fMRI functional connectivity detection.

Multivariate Mathematical Morphology Multivariate Mathematical Morphology aims to extend the definitions and results of Mathematical Morphology which have proven successful in gray scale images to images whose pixels are high dimensional vectors, such as functional Magnetic Resonance Images. A fundamental issue in Multivariate Mathematical Morphology is the definition of a convenient ordering over the multivariate data space ensuring that ensuing morphological operators and filters are mathematically consistent. A recent technique consists in

[1]The original works were devoted to unsupervised hyperspectral image segmentation, therefore the use of the name *endmember* for the selected vectors.

using the outputs of two-class classifiers trained on the data to build meaningful reduced orderings. These two classes are defined as background and foreground classes corresponding to target and background features of the data. We have introduced several approaches to define reduced supervised orderings based on the recall error of the Lattice Auto-Associative Memories (LAAM), measured by the Chebyshev distance. Foreground supervised orderings use one set of training data from a foreground class, whilst background/foreground supervised orderings use two training data sets, one for each relevant class. We work on the definition of Lattice Computing approach to identify functional networks in resting state fMRI data (rsfMRI) looking for biomarkers of cognitive or neurodegenerative diseases. We provide results showing that the approach may allow to find brain networks with quite different connectivities that are even amenable to supervised classification experiments.

1.3 Publications

1. Darya Chyzhyk, Manuel Graña, "Results on a Lattice Computing based group analysis of schizophrenic patients on resting state fMRI", In J.M. Ferrandez et al. (Eds.), editor, IWINAC 2013, Part II, volume 7931 of LNCS, pages 131-139. Springer, 2013.

2. Maite Termenon, Darya Chyzhyk, Manuel Graña, Alberto Barros-loscertales, Cesar Avila, "Cocaine Dependent Classification on MRI Data extracting Features from Voxel Based Morphometry", In J.M. Ferrandez et al. (Eds.), editor, IWINAC 2013, Part II, volume 7931 of LNCS, pages 140-148. Springer, 2013.

3. Darya Chyzhyk, Borja Ayerdi, Josu Maiora, "Active Learning with Bootstrapped Dendritic Classifier applied to medical image segmentation", Pattern Recognition Letters (in press).

4. Darya Chyzhyk, Manuel Graña, "Bootstrapped Dendritic Classifiers in MRI analysis for Alzheimer's Disease recognition", KES, chapter.

5. Gerhard X. Ritter, Darya Chyzhyk, Gonzalo Urcid, Manuel Graña, "A Novel Lattice Associative Memory Based on Dendritic Computing", Hybrid Artificial Intelligent Systems in: HAIS'2012 Proceedings of the 7th international

conference on Hybrid Artificial Intelligent Systems, Volume Part II, pp. 491-502, 2012. LNCS 7209.

6. Darya Chyzhyk, Alexandre Savio and Manuel Graña, "Evolutionary ELM wrapper feature selection for Alzheimer's disease CAD on anatomical brain MRI", ELM 2012, Neurocomputing (in press).

7. Darya Chyzhyk, "Bootstrapped Dendritic Classifiers for Alzheimer's Disease classification on MRI features", Advances in Knowledge-Based and Intelligent Information and Engineering Systems. Frontiers in Artificial Intelligence and Applications (FAIA) series, Vol. 243, pages 2251 - 2258, 2012. Eds: Manuel Graña, Carlos Toro, Jorge Posada, Robert J. Howlett and Lakhmi C. Jain.

8. Manuel Graña, Darya Chyzhyk, "Hybrid multivariate morphology using lattice auto-associative memories for resting-state fMRI network discovery", HIS 2012, Pune India, 4-7 december 2012, Published in: 12th International Conference on Hybrid Intelligent Systems, , 537-542, IEEE 2012.

9. Darya Chyzhyk, Manuel Graña, Alexandre Savio, Josu Maiora, "Hybrid Dendritic Computing with Kernel-LICA applied to Alzheimer's Disease detection in MRI", Neurocomputing, 2012, Volumen 75, issue 1, pp. 72-77.

10. Darya Chyzhyk, Ann K. Shinn, and Manuel Graña, "Exploration of LICA Detections in Resting State fMRI", 4th. International work-Conference on the Interplay between Natural and Artificial Computation (IWINAC 2011), La Palma, Canary Islands, Spain. 30 May - 3 June, 2011. Published in: IWINAC'11 Proceedings of the 4th international conference on Interplay between natural and artificial computation: new challenges on bioinspired applications - Volume Part II, pp.104-111.

11. Darya Chyzhyk, Manuel Graña, "Optimal Hyperbox shrinking in Dendritic Computing applied to Alzheimer's Disease detection in MRI", Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2011), Salamanca, Spain. 6-8th April, 2011. Advances in Intelligent and Soft Computing, 2011, Volume 87/2011, 543-550, Editors: Emilio Corchado, Václav Snášel, Javier Sedano, Aboul Ella Hassanien, José Luis Calvo.

12. Darya Chyzhyk, Maite Termenon, and Alexandre Savio, "A Comparison of VBM results by SPM, ICA and LICA", Hybrid Artificial Intelligent Systems, Part II, Emilio S. Corchado Rodriguez; Manuel Grana Romay; Alexandre Manhaes Savio (Eds.), (LNCS/LNAI) Vol. 6077, Springer-Verlag Berlin Heidelberg, 2010, pp. 429 - 435 Vol. 2

13. Manuel Graña, Darya Chyzhyk, Maite García-Sebastián, Carmen Hernández, "Lattice Independent Component Analysis for functional Magnetic Resonance Imaging", Information Sciences 181, issue 10, 2011, pp. 1910-1928.

14. Alexandre Savio, Maria T Garcia-Sebastian, Darya Chyzhyk, Carmen Hernandez, Manuel Graña, Andone Sistiaga, Adolfo Lopez de Munain, Jorge Villanua, "Neurocognitive disorder detection based on Feature Vectors extracted from VBM analysis of structural MRI", Computers in Biology and Medicine 41 (2011), pp. 600-610.

15. Alexandre Savio, Maite García-Sebastián, Andone Sistiaga, Darya Chyzhyk, Esther Fernández, Fermín Moreno, Elsa Fernández, Manuel Graña, Jorge Villanúa, Adolfo López de Munain, "Machine Learning Approach for Myotonic Dystrophy Diagnostic Support from MRI", in Juan M. Górriz, Elmar W. Lang and Javier Ramírez (Eds), Recent Advances in Biomedical Signal Processing, 93 - 101 2010 Bentham Science Publishers.

16. Darya Chyzhyk and Alexandre Savio, "Feature extraction from structural MRI images based on VBM: data from OASIS database", Technical Report GIC-UPV-EHU-RR-2010-10-14, Grupo de Inteligencia Computacional UPV/EHU, 2010

17. Darya Chyzhyk, "Inferencia en FSL para morfometria basada en vóxel (voxel base morphometry)", III Jornada de Inteligencia Computacional, JIC'09. Servicio editorial de la UPV/EHU, 2009

1.4 Contents of the Thesis

The contents of the Thesis are organized as follows:

1. Chapter 2 provides a review of Lattice Computing fundamentals that are being used in the remainder of the Thesis, mostly basic results about Lattice Auto-associative Memories and Dendritic Computing approaches.

2. Chapter 3 provides results of Dendritic Computing approaches on a variety of applications. Specifically, we discuss several enhancements of the single lattice neurong with dendritic computing by introducing shrinking hyperboxes, kernel LICA preprocessing and ensembles. The methods are applied to Alzheimer's Disease classification and CTA image segmentaion. Finally, a novel autoassociative memory is proposed.

3. Chapter 4 reports results on the application of LICA on three case studies: Voxel Based Morphometry (VBM) of a subset of the OASIS database, a selection of synthetic fMRI data, and a resting state fMRI database also described in Appendix A.

4. Chapter 5 provides some results on Multivariate Mathematical Morphology applied to resting state fMRI. Specifically, the work tries to find some image biomarkers for Schizophrenia patients, with and without auditory hallucinations. Multivariate Mathematical Morphology is built on top of Lattice Auto-associative Memory recall, as a Lattice Computing extension to the supervised reduced ordering approaches.

5. Chapter 6 recalls some conclusions from the cross section of the contents of the Thesis.

6. Appendix A provides the description of the databases that are used for the computational experiments along the Thesis, specifically, the Alzheimer's Disease, Schizophrenia and Abdominal Aortic Aneurysm databases, which exemplify diverse image modalities. It also provides some introduction to the diverse data modalities, and some techniques such as VBM.

Chapter 2

Lattice Computing fundamentals

In this Chapter we gather Lattice Computing miscelanea definitions relevant to various chapters in the Thesis. The main aim is to provide a short review of definitions and results that would allow the smooth reading of ensuing chapters, identifying the main contributions of the Thesis. The contents of the Chapter are as follows: Section 2.1 provides basic definitions from Lattice Theory. Section 2.2 provides a review of Lattice Associative Memories, with an emphasis in the definition of Lattice Independence which is relevant for subsequent sections. Section 2.3 provides the basic definitions of Dentritic Computing classifier systems. Section 2.4 introduces the Lattice Independent Component Analysis (LICA). Section 2.5 provides the description of an endmember induction algorithm used in some of the experiments in the Thesis.

2.1 Lattice Theory

George Davis Birkhoff along with his son, Garrett Birkhoff, made a long standing major contribution to the development of Lattice Theory [**?**]. Their articles and books have been in its time the encyclopedia of Lattice Theory, containing the building block definitions as well as reflecting relations with different fields of mathematics and other sciences. The richness and diversity of their ideas have certainly contributed to our ability to look at the science on the other side stimulating the application of lattice theory to solve real life problems. In this review chapter, we use actual notation and definitions from [**?**] [**?**] [**?**].

Definition 1. A relation $\leq$ on a set X is called a *partial order* on X if and only

if for every $x, y, z \in X$ the reflexivity, antisymmetry and transitivity are satisfied Table 2.1.

A partially ordered is totally ordered set if it satisfies linearity property.

Definition 2. A *lattice* is a partially ordered set L such that for any two elements $x, y, \in L$, $inf\{x,y\}$ and $sup\{x,y\}$ exist. If L is a lattice, then we define $x \wedge y = inf\{x,y\}$ and $y \wedge x = sup\{x,y\}$.

A sublattice of a lattice L is a subset X of L such that for each pair $x, y, \in X$ we have that $x \wedge y \in X$ and $y \wedge x \in X$.

A lattice L is sais to be complete if and only if for each of its subset X $inf\{x,y\}$ and $sup\{x,y\}$ exist. We define the symbols $\bigwedge X = inf X$ and $\bigvee X = sup X$.

Lemma 3. *Let P be a partially ordered set in wich $\bigvee H \bigwedge H$* exist for all $H \subseteq P$, *Then P is a complete lattice.*

Let $H \subseteq X$, $x \in X$. Than x is an upper bound of H, if $h \leq a$ for all $h \in H$. And upper bound x of H is the least upper bound of H or supremum of H if, for any upper bound y of H, we have $x \leq y$. We shall write $x = sup\{x,y\}$, or $x = \bigvee X$. The concept of lower bound and greatest lower bound or infinum are similarity defined; the latter is denoted by $inf H$ or $\bigwedge H$. We will use this notation $x \wedge y = inf\{x,y\}$ and $y \wedge x = sup\{x,y\}$ and also call $\vee$ join and $\wedge$ meet. In latteces, they are both binary operations, which means that they can be applied to a pair of elements $x, y, \in L$ to produce again the element of L. It should be noted that they are idempotent, commutative and associative Table 2.1.

Lattices have special elements called the top $\top$ and bottom $\bot$. The top is the unique greatest upper bound, whereas the bottom is the unique least lower bound.

2.2 Lattice Associative Memories

The concept of an associative memory is a fairly intuitive one as it is based on the observation that an associative memory seems to be one of the primary functions of the brain. We easily *associate* the face of a friend with that of the friend's name, or a name with a telephone number. For this reason artificial neural networks (ANNs) that are capabable of storing several types of patterns and corresponding associations are referred to as *associative memories*. Such memories retrieve stored associations when presented with corresponding input patterns. An associative memory

Ordering	
Order	$\leq$
Reflexivity	For all x, $x \leq x$
Antisymmetry	if $x \leq y$ and $y \leq x$ then $x = y$
Transitivity	if $x \leq y$ and $y \leq z$ then $x \leq z$
Linearity	For all x, y, $x \leq y$, $y \leq x$

Operations	
Join	$\vee$
Meet	$\wedge$
Idempotency	$x \vee x = x$
	$x \wedge x = x$
Commutativity	$x \vee y = y \vee x$
	$x \wedge y = y \wedge x$
Associativity	$(x \vee y) \vee z = x \vee (y \vee z)$
	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$
Absorption	$x \wedge (x \vee y) = x \vee (x \wedge y) = x$
Distributivity	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
	$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$
Translation	if $x \leq y$, then $a + x + b \leq a + y + b$

Table 2.1: Lattice distinctive feature

is said to be *robust in the presence of noise* if presented with a corrupted version of a prototype input pattern it is still capable of retrieving the correct association.

The work on Lattice Associative Memories (LAM) stems from the consideration of the bounded lattice ordered group (blog) $(\mathbb{R}_{\pm\infty}, \vee, \wedge, +, +')$ as the alternative to the algebraic field $(\mathbb{R}, +, \cdot)$ commonly used for the definition of Neural Network algorithms. Here $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}_{\pm\infty}$ the extended real numbers, $\wedge$ and $\vee$ denote, respectively, the binary max and min operations, and $+$, $+'$ denote addition and its dual operation. In our current context addition is self-dual. If $x \in \mathbb{R}_{\pm\infty}$, then its additive conjugate is $x^* = -x$. For a matrix $A \in \mathbb{R}_{\pm\infty}^{n\times m}$ its conjugate matrix is given by $A^* \in \mathbb{R}_{\pm\infty}^{n\times m}$, where each entry $a_{ij}^* = [A^*]_{ij}$ is given by $a_{ij}^* = (a_{ji})^*$.

The LAM were first introduced in [**?**, **?**] as Morphological Associative Memories, a name still used in recent publications [**?**], but we follow the terminology introduced in [**?**, **?**] in the general framework of Lattice Computing.

2.2.1 Definitions

Given a set of input/output pattern pairs $(X,Y) = \left\{ \left(\mathbf{x}^\xi, \mathbf{y}^\xi\right); \xi = 1,..,k \right\}$, a linear heteroassociative neural network based on the pattern's cross correlation is built up as

$$W = \sum_\xi \mathbf{y}^\xi \cdot \left(\mathbf{x}^\xi\right)'. \tag{2.1}$$

Mimicking this constructive procedure [**?**, **?**] proposed the following constructions of Lattice Heteroassociative Memories (LHAM):

$$W_{XY} = \bigwedge_{\xi=1}^{k} \left[\mathbf{y}^\xi \times \left(-\mathbf{x}^\xi\right)'\right] \text{ and } M_{XY} = \bigvee_{\xi=1}^{k} \left[\mathbf{y}^\xi \times \left(-\mathbf{x}^\xi\right)'\right], \tag{2.2}$$

where $\times$ is any of the $\boxtimes$ or $\boxtimes$ operators. Here $\boxtimes$ and $\boxtimes$ denote the max and min matrix product [**?**, **?**]. respectively defined as follows:

$$C = A \boxtimes B = [c_{ij}] \Leftrightarrow c_{ij} = \bigvee_{k=1,\ldots,n} \left\{a_{ik} + b_{kj}\right\}, \tag{2.3}$$

$$C = A \boxtimes B = [c_{ij}] \Leftrightarrow c_{ij} = \bigwedge_{k=1,\ldots,n} \left\{a_{ik} + b_{kj}\right\}. \tag{2.4}$$

If $X = Y$ then the LHAM memories are Lattice Autoassociative Memories

(LAAM). Conditions of perfect recall by the LHAM and LAAM of the stored patterns proved in [**?**, **?**] encouraged research. In the continuous case, the LAAM is able to store and recall any set of patterns, that is: $W_{XX} \boxtriangledown X = X = M_{XX} \boxtriangle X$ holds for any continuous X. However, this result holds when we deal with noise-free patterns. Research on robust recall [**?**, **?**, **?**] based on the so-called kernel patterns lead to the notion of morphological independence, in the erosive and dilative sense, and finally to the definition of Lattice Independence and Strong Lattice Independence.

2.2.2 Lattice Independence

Definition 4. Given a set of vectors $X = \left\{\mathbf{x}^1, ..., \mathbf{x}^k\right\} \subset \mathbb{R}^n$ a *linear minimax combination* of vectors from this set is any vector $\mathbf{x} \in \mathbb{R}^n_{\pm\infty}$ which is a *linear minimax sum* of these vectors:

$$x = \mathscr{L}\left(\mathbf{x}^1, ..., \mathbf{x}^k\right) = \bigvee_{j \in J} \bigwedge_{\xi=1}^{k} \left(a_{\xi j} + \mathbf{x}^{\xi}\right),$$

where J is a finite set of indices and $a_{\xi j} \in \mathbb{R}_{\pm\infty}$ $\forall j \in J$ and $\forall \xi = 1, ..., k$.

The *linear minimax span* of vectors $\left\{\mathbf{x}^1, ..., \mathbf{x}^k\right\} = X \subset \mathbb{R}^n$ is the set of all linear minimax sums of subsets of X, denoted $LMS\left(\mathbf{x}^1, ..., \mathbf{x}^k\right)$.

Now we are able to define the notions of lattice dependence.

Definition 5. Given a set of vectors $X = \left\{\mathbf{x}^1, ..., \mathbf{x}^k\right\} \subset \mathbb{R}^n$, a vector $\mathbf{x} \in \mathbb{R}^n_{\pm\infty}$ is *lattice dependent* if and only if $x \in LMS\left(\mathbf{x}^1, ..., \mathbf{x}^k\right)$. The vector $\mathbf{x}$ is *Lattice Independent* if and only if it is not lattice dependent on X. The set X is said to be *Lattice Independent* if and only if $\forall \lambda \in \{1, ..., k\}$, $\mathbf{x}^{\lambda}$ is Lattice Independent of $X \setminus \left\{\mathbf{x}^{\lambda}\right\} = \left\{\mathbf{x}^{\xi} \in X : \xi \neq \lambda\right\}$.

The definition of Lattice Independence supersedes earlier definitions [**?**] of erosive and dilative morphological independence. This definition of lattice dependence is closely tied to the study of the LAAM fixed points when they are interpreted as lattice transformations, as stated by the following theorem.

The following definition introduces a matrix property needed to refine the idea of lattice independence.

Definition 6. A set of vectors $X = \left\{\mathbf{x}^1, ..., \mathbf{x}^k\right\} \subset \mathbb{R}^n$ is said to be *max dominant* if

and only if for every $\lambda \in \{1,...,k\}$ there exists and index $j_\lambda \in \{1,...,n\}$ such that

$$x^\lambda_{j_\lambda} - x^\lambda_i = \bigvee_{\xi=1}^{k} \left(x^\xi_{j_\lambda} - x^\xi_i \right) \forall i \in \{1,...,n\}.$$

Similarly, X is said to be *min dominant* if and only if for every $\lambda \in \{1,...,k\}$ there exists and index $j_\lambda \in \{1,...,n\}$ such that

$$x^\lambda_{j_\lambda} - x^\lambda_i = \bigwedge_{\xi=1}^{k} \left(x^\xi_{j_\lambda} - x^\xi_i \right) \forall i \in \{1,...,n\}.$$

The expressions that compound this definition appeared in the early theorems about perfect recall of Morphological Associative Memories [**?**, **?**]. Their value as an identifiable property of the data has been discovered in the context of the formalization of the relationship between Strong Lattice Independence, defined below, and the Affine Independence of classical linear analysis.

Definition 7. A set of Lattice Independent vectors $\left\{\mathbf{x}^1,...,\mathbf{x}^k\right\} \subset \mathbb{R}^n$ is said to be *Strongly Lattice Independent* (SLI) if and only if X is max dominant or min dominant or both.

As said before, min and max dominance are the conditions for perfect recall in LAM. Per construction, the column vectors of LAAM are diagonally min or max dominant, depending of their erosive or dilative nature, therefore they will be SLI, *if* they are Lattice Independent. We recall that a set of vectors $X = \{\mathbf{x}_1,\ldots,\mathbf{x}_M\}$ is said to be Linearly Independent if the unique solution to the equation $\sum_{i=1}^{M} a_i\mathbf{x}_i = \mathbf{0}$ is given by $a_i = 0$ for all $i \in \{1,\ldots,M\}$. A set X is an Affine Independent set if the solution to the simultaneous equations $\sum_{i=1}^{M} a_i\mathbf{x}_i = \mathbf{0}$ and $\sum_{i=1}^{M} a_i = \mathbf{0}$ is given by $a_i = 0$ for all $i \in \{1,\ldots,M\}$. Therefore, Linear Independence is a necessary condition for Affine Independence but not viceversa. A set of M Affine Independent points defines a simplex, so that if we find it then we can assume the Linear Mixing Model to perform data unmixing, such as done in Section 2.4.

Conjecture 8. *[**?**, **?**] If $X = \left\{\mathbf{x}^1,...,\mathbf{x}^k\right\} \subset \mathbb{R}^n$ is SLI then X is Affine Independent.*

This conjecture (stated as theorem in [**?**, **?**]) is the key result relating the linear analysis based on the Linear Mixing Model and the non-linear lattice analysis. If true, it means that the construction of the LAAM provides the starting point for

obtaining sets of Affine Independent vectors that could be used as Lattice Independent Sources for the unmixing algorithms as done in the Lattice Independent Component Analysis of Section 2.4.

Theorem 9. *[?] Let $X = \left\{\mathbf{x}^1, ..., \mathbf{x}^k\right\} \subset \mathbb{R}^n$ and let W and its dual M be the set of vectors consisting of the columns of the matrix W_{XX} and its dual M_{XX}. Let $F(X)$ denote the set of fixed points of the LAAM constructed from set X. There exist $V \subset W$ and $N \subset M$ such that V and N are SLI and $F(X) = F(V) = F(N)$ or, equivalently, $W_{XX} = W_{VV}$ and $M_{XX} = M_{NN}$.*

The key idea of this theorem is that it is possible to built a set of SLI vectors from the column vectors of a LAAM. Taking into account that the column vectors of a LAAM are diagonally max or min dominant (depending on the kind of LAAM), it suffices to find a subset which is lattice independent. It also uses the fact that a subset of a set of max or min dominant vectors is also min or max dominant. The constructive proof of the theorem provides algorithms to find these sets of SLI. It removes iteratively the detected lattice dependent column vectors. Detection lies in the fact that $W_{XX} = W_{WW} = W_{VV}$ and $M_{XX} = M_{MM} = M_{NN}$ when the vectors removed from W or M to obtain V or N are lattice dependent on the remaining ones.

2.2.3 Perfect recall

The following theorems about conditions for LAM and LAAM perfect recall were estabilised in [**?**]

Theorem 10. *Let (X,Y) denote the associate sets of pattern vectors pairs. Whenever exist perfect recall memories A and B such that $A \boxtimes \mathbf{x}^\xi = \mathbf{y}^\xi$ and $B \boxtimes \mathbf{x}^\xi = \mathbf{y}^\xi$ for $\xi = 1, ..., k$, then*

$A \leq W_{XY} \leq M_{XY}$ and $\forall \xi$, $W_{XY} \boxtimes \mathbf{x}^\xi = \mathbf{y}^\xi = M_{XY} \boxtimes \mathbf{x}^\xi$.

Here W_{XY} is the least upper bound of all perfect recall memories involving the $\boxtimes$ operation and M_{XY} is the greatest lower bound of all perfect recall memories involving the $\boxtimes$ operation.

Theorem 11. *W_{XY} is a perfect recall memory for the pattern association $\left(\mathbf{x}^\lambda, \mathbf{y}^\lambda\right)$* if and only if each row of the matrix $\left[\mathbf{y}^\lambda + \left(\mathbf{x}^\lambda\right)^*\right] - W_{XY}$ contains a zero entry. Similary, *M_{XY} is a perfect recall memory for the pattern* pair $\left(\mathbf{x}^\lambda, \mathbf{y}^\lambda\right)$ if and only if each row of the matrix $M_{XY} - \left[\mathbf{y}^\lambda + \left(\mathbf{x}^\lambda\right)^*\right]$ contains a zero entry.

If $X = Y$ then W_{XX} and M_{XX} are called Lattice Auto-Associative Memories (LAAM). Since:

$$w_{ii} = \bigwedge_{\xi=1}^{k} \left(x_i^\xi - x_i^\xi \right) = 0 = \bigvee_{\xi=1}^{k} \left(x_i^\xi - x_i^\xi \right) = m_{ii},$$

the diagonals of the memories W_{XX} and M_{XX} consist complitelly of zeros. The following Corollary is concluded from Theorem 11.

Corollary 12. *$W_{XY} \boxtimes \mathbf{x}^\xi = \mathbf{x}^\xi = M_{XY} \boxtimes \mathbf{x}^\xi$ for each $\xi = 1, \ldots, k$.*

Therefore, if we can obtain an erosive LAM with perfect recall, then there is a corresponding dilative LAM that has perfect recall property.

2.2.4 Lattice Approximation

The natural way to deal with approximation in the Lattice Computing framework is to use the Chebyshev distance [**?**] given by the greatest componentwise absolute difference between two vectors, it is denoted $\varsigma(\mathbf{x}, \mathbf{y})$ and can be computed as follows: $\varsigma(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^* \boxtimes \mathbf{y}) \vee (\mathbf{y}^* \boxtimes \mathbf{x})$. The Chebyshev-best approximation of $\mathbf{c}$ by $f(\mathbf{x})$ subject to $\mathbf{x} \in S$, is the minimization of $\varsigma(f(\mathbf{x}), \mathbf{c})$ subject to $\mathbf{x} \in S$.

Theorem 13. *[**?**] Given $B \in \mathbb{R}^{m \times n}$ and $\mathbf{c} \in \mathbb{R}^m$, a Chebyshev-best solution to the approximation of $\mathbf{c}$ by $B \boxtimes \mathbf{x}$ subject to the constraint $B \boxtimes \mathbf{x} < \mathbf{c}$ is given by $\mathbf{x}^\# = B^* \boxtimes \mathbf{c}$ and $\mathbf{x}^\#$ is the greatest such solution.*

In our incremental algorithm ILSIA we will need to solve the unconstrained minimization problem

$$\min \varsigma(B \boxtimes \mathbf{x}, \mathbf{c}),$$

in order to decide if the input vector is already well approximated by a fixed point of the LAAM constructed from the selected enmembers.

Theorem 14. *[**?**]Given $B \in \mathbb{R}^{m \times n}$ and $\mathbf{c} \in \mathbb{R}^m$, a Chebyshev-best solution to the approximation of $\mathbf{c}$ by $B \boxtimes \mathbf{x}$ is given by $\mu + \mathbf{x}^\#$ where μ is such that $2\mu = \varsigma\left(B \boxtimes \mathbf{x}^\#, \mathbf{c}\right) = \left(B \boxtimes \mathbf{x}^\#\right)^* \boxtimes \mathbf{c}$.*

This theorem has resulted in enhanced robust recall for LAAM under general noise conditions, compared with other Associative Memories proposed in the literature. It has also been applied to produce a lattice based nearest neighbor classification scheme with good results on standard benchmark classification problems.

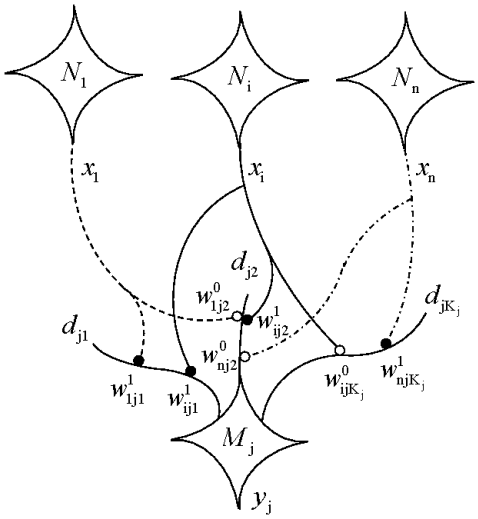


Figure 2.1: Terminal branches of axonal fibers originating from the presynaptic neurons make contact with synaptic sites on dendritic branches of M_j.

In fact, we use this approximation for the definition of the LAAM-based reduced ordering that provides a Lattice Computing based Multivariate Mathematical Morphology in Chapter 5.

2.3 Dendritic computing

First we recall a very general definition of Dendritic Lattice models of ANN, of which the single layer dendritic network is a salient example, due to its simplicity, which is introduced next.

2.3.1 The Dendritic Lattice Based Model of ANNs

Roughly speaking, a lattice based neural network is an ANN in which the basic neural computations are based on the operations of a lattice ordered group. By a lattice ordered group we mean a set L with an associated algebraic structure $(L, \vee, \wedge, +)$, where $(L, \vee, \wedge)$ is a lattice and $(L, +)$ is a group with the property that every group translation is isotone Table 2.1 . Given the set $\mathfrak{O} = \{\vee, \wedge, +\}$ of lattice goup operations, then the symbols $\oplus$ and $\otimes$ will mean that $\oplus, \otimes \in \mathfrak{O}$ but are

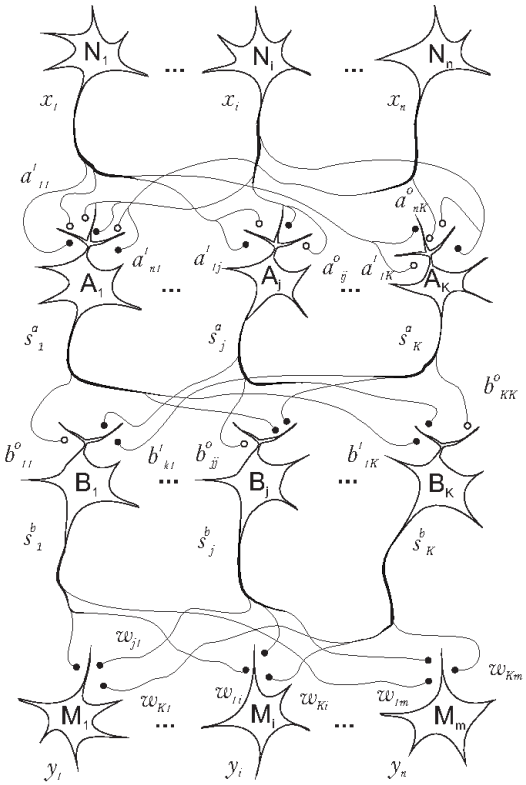


Figure 2.2: General multi-layer structure of a dendritic network

not explicitly specified lattice operations. Similarly, symbols of form $\bigoplus$ and $\bigotimes$ will denote lattice operations derived from the operations $\oplus$ and $\otimes$, respectively. For example, $\bigoplus_{i=1}^{n} a_i = a_1 \oplus a_2 \oplus \cdots \oplus a_n$. Hence, specifying $\oplus = \vee$, then $\bigoplus_{i=1}^{n} a_i = \bigvee_{i=1}^{n} a_i = a_1 \vee a_2 \vee \cdots \vee a_n$.

In the dendritic model of ANNs, a finite set of presynaptic neurons $N_1, \ldots, N_n$ provides information through its axonal arborization to the dendritic trees of some other finite set of postsynaptic neurons $M_1, \ldots, M_m$. The dendritic tree of a postsynaptic neuron M_j is assumed to consist of a finite number of branches $d_{j1}, \ldots, d_{jK_j}$ which contain the synaptic sites upon which the axonal fibers of the presynaptic neurons terminate. The *strength* of the synapse on the kth dendritic branch d_{jk} ($k \in \{1, \ldots, K(j)\}$) which serves as a synaptic site for a terminal axonal branch fiber of N_i is denoted by w_{ijk}^{ℓ} and is also called its *synaptic weight*. The superscript ℓ is associated with the postsynaptic response that is generated within and in close

proximity of the synapse. Specifically, $\ell = 0$ and $\ell = 1$ denote an inhibitory or excitatory postsynaptic response, respectively. It is possible for several axonal fibers to synapse on the same or different synaptic sites on a given branch d_{jk}, with the former case implying that $w^{\ell}_{ijk} = w^{\ell}_{hjk}$. The total response (or output) of d_{jk} to the received input at its synaptic sites is given by

$$\tau^j_k(\mathbf{x}) = p_{jk} \bigoplus_{i \in I(k)} \bigotimes_{\ell \in \mathscr{L}(i)} [(-1)^{1-\ell} \left(x_i + w^{\ell}_{ijk} \right)], \tag{2.5}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in L^n$ with L^n denoting the n-fold cartesian product of L, $x_i \in L$ denotes the information propagated by N_i via its axon and axonal branches, $\mathscr{L}(i) \subseteq \{0,1\}$ corresponds to the postsynaptic response generated at the synaptic region to the input received from N_i, and $I(k) \subseteq \{1, \ldots, n\}$ corresponds to the set of all presynaptic neurons with terminal axonal fibers that synapse on the kth dendritic branch of M_j. The value $p_{jk} \in \{-1, 1\}$ marks the final signal outflow from the kth branch as inhibitory if $p_{jk} = -1$ and excitatory if $p_{jk} = 1$. The value $\tau^j_k(\mathbf{x})$ is passed to the cell body of M_j and the state of M_j is a function of the combined values received from its dendritic structure and is given by

$$\tau^j(\mathbf{x}) = p_j \bigotimes_{k=1}^{K_j} \tau^j_k(\mathbf{x}), \tag{2.6}$$

where K_j denotes the total number of dendritic branches of M_j and $p_j = \pm 1$ denotes the response of the cell to the received input. Here again $p_j = -1$ means rejection (inhibition) and $p_j = 1$ means acceptance (exitation) of the received input. Figure 2.3.1 illustrates the neural pathways from the presynaptic neurons to the postsynaptic neuron M_j. Figure 2.3.1 illustrates a dendritic network.

2.3.2 Dendritic computing on a single layer

The prime example of a lattice ordered group is the set $\mathbb{R}$ of real numbers together with the binary operations of the maximum ($\vee$) and minimum ($\wedge$) of two numbers and the group operation of addition, denoted by $(\mathbb{R}, \vee, \wedge, +)$. A single layer morphological neuron endowed with dendrite computation based on lattice algebra was introduced in [**?**]. Figure 2.3 illustrates the structure of a single output class single layer Dendritic Computing system, where D_j denotes the dendrite with associated inhibitory and excitatory weights $\left(w^0_{ij}, w^1_{ij} \right)$ from the synapses coming from the

Algorithm 2.1 Dendritic Computing learning algorithm based on elimination.

Training set $T = \left\{\left(\mathbf{x}^{\xi}, c_{\xi}\right) \mathbf{x}^{\xi} \in \mathbb{R}^n, c_{\xi} \in \{0,1\}; \xi = 1,\ldots,m\right\}$, $C_1 = \left\{\xi : c_{\xi} = 1\right\}, C_0 = \left\{\xi : c_{\xi} = 0\right\}$

1. Initialize $j = 1, I_j = \{1,\ldots n\}, P_j = \{1,\ldots,m\}, L_{ij} = \{0,1\}$,

$$w^1_{ij} = -\bigwedge_{c_{\xi}=1} x^{\xi}_i; w^0_{ij} = -\bigvee_{c_{\xi}=1} x^{\xi}_i, \textit{biologically} \forall i \in I$$

2. Compute response of the current dendrite D_j, with $p_j = (-1)^{\text{sgn}(j-1)}$:

$$\tau_j\left(\mathbf{x}^{\xi}\right) = p_j \bigwedge_{i \in I_j} \bigwedge_{l \in L_{ij}} (-1)^{1-l}\left(x^{\xi}_i + w^l_{ij}\right), \forall \xi \in P_j.$$

3. Compute the total response of the neuron:

$$\tau\left(\mathbf{x}^{\xi}\right) = \bigwedge_{k=1}^{j} \tau_k\left(\mathbf{x}^{\xi}\right); \xi = 1,\ldots,m.$$

4. If $\forall \xi \left(f\left(\tau\left(\mathbf{x}^{\xi}\right)\right) = c_{\xi}\right)$ the algorithm stops here with perfect classification of the training set.

5. Create a new dendrite $j = j+1, I_j = I' = X = E = H = \emptyset, D = C_1$

6. Select $\mathbf{x}^{\gamma}$ such that $c_{\gamma} = 0$ and $f\left(\tau\left(\mathbf{x}^{\gamma}\right)\right) = 1$.

7. $\mu = \bigwedge_{\xi \neq \gamma}\left\{\bigvee_{i=1}^{n}\left|x^{\gamma}_i - x^{\xi}_i\right| : \xi \in D\right\}$.

8. $I' = \left\{i : \left|x^{\gamma}_i - x^{\xi}_i\right| = \mu, \xi \in D\right\}; X = \left\{\left(i, x^{\xi}_i\right) : \left|x^{\gamma}_i - x^{\xi}_i\right| = \mu, \xi \in D\right\}$.

9. $\forall \left(i, x^{\xi}_i\right) \in X$

   (a) if $x^{\gamma}_i > x^{\xi}_i$ then $w^1_{ij} = -(x^{\xi}_i + \mu), E_{ij} = \{1\}$

   (b) if $x^{\gamma}_i < x^{\xi}_i$ then $w^0_{ij} = -(x^{\xi}_i - \mu), H_{ij} = \{0\}$

10. $I_j = I_j \bigcup I'; L_{ij} = E_{ij} \bigcup H_{ij}$

11. $D' = \left\{\xi \in D : \forall i \in I_j, -w^1_{ij} < x^{\xi}_i < -w^0_{ij}\right\}$. If $D' = \emptyset$ then goto step 2, else $D = D'$ goto step 7.

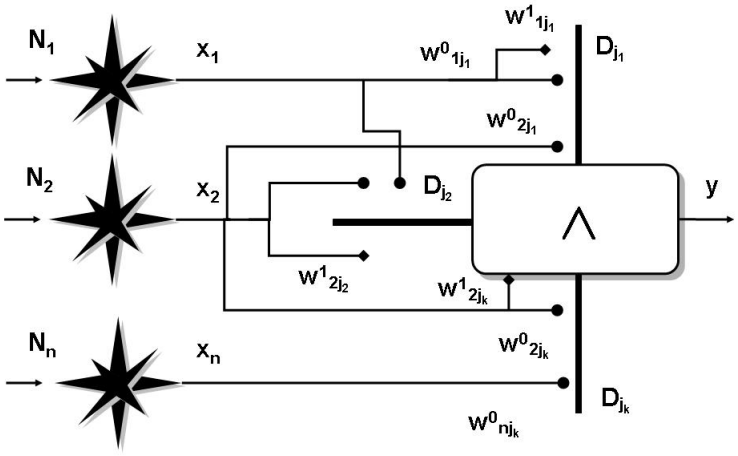


Figure 2.3: A single output single layer Dendritic Computing system.

i-th input neuron. It is the meaning of Dendritic Computing classifier all across the Thesis. Thus, for example, eqn.2.5 could assume the form

$$\tau_k^j(\mathbf{x}) = p_{jk} \bigvee_{i \in I(k)} \bigwedge_{\ell \in \mathscr{L}(i)} (-1)^{1-\ell} \left(x_i + w_{ijk}^{\ell} \right), \tag{2.7}$$

where $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, and $x_i \in \mathbb{R}$, while eqn.2.6 could be of form

$$\tau^j(\mathbf{x}) = p_j \sum_{k=1}^{K_j} \tau_k^j(\mathbf{x}). \tag{2.8}$$

Alternatively, the response of the j-th dendrite may be specified as follows:

$$\tau_j\left(\mathbf{x}^{\xi}\right) = p_j \bigwedge_{i \in I_j} \bigwedge_{l \in L_{ij}} (-1)^{1-l} \left(x_i^{\xi} + w_{ij}^{l} \right), \tag{2.9}$$

where $l \in L \subseteq \{0,1\}$ identifies the existence and inhibitory/excitatory character of the weight, $L_{ij} = \emptyset$ means that there is no synapse from the i-th input neuron to the j-th dendrite; $p_j \in \{-1,1\}$ encodes the inhibitory/excitatory response of the dendrite. It has been shown [**?**] that these dendritic models have powerful

approximation properties. The total response of the neuron is given by:

$$\tau(x_i) = f\left(\bigwedge_{j=1}^{J} \tau_j(x_i)\right),$$

where $f(x)$ is the Heaviside hardlimiter function. A constructive algorithm obtains perfect classification of the train dataset using J dendrites.

In fact, [**?**] showed that this model is able to approximate any compact region in higher dimensional Euclidean space to within any desired degree of accuracy. They provide a constructive algorithm which is the basis for the present paper. The hard-limiter function of step 3 is the signum function. The algorithm starts building a hyperbox enclosing all pattern samples of class 1, that is, $C_1 = \left\{\xi : c_\xi = 1\right\}$. Then, the dendrites are added to the structure trying to remove misclasified patterns of class 0 that fall inside this hyperbox. In step 6 the algorithm selects at random one such misclassified patterns, computes the minimum Chebyshev distance to a class 1 pattern and uses the patterns that are at this distance from the misclassified pattern to build a hyperbox that is removed from the C_1 initial hyperbox. In this process, if one of the bounds is not defined, $L_{ij} \neq \{0,1\}$, then the box spans to infinity in this dimension. One of the recent improvements [**?**] consists in considering rotations of the patterns obtained from some learning process. Then, the response of the dendrite is given by:

$$\tau_j\left(\mathbf{x}^\xi\right) = p_j \bigwedge_{i \in I_j} \bigwedge_{l \in L_{ij}} (-1)^{1-l}\left(R\left(\mathbf{x}^\xi\right)_i + w_{ij}^l\right),$$

where R denotes the rotation matrix. The process of estimating R can be very time consuming, it is a local process performed during steps 7 to 10 of the learning process of algorithm 2.1.

2.4 Lattice Independent Component Analysis (LICA)

In the domain of remote sensing hyperspectral image processing, a Linear Mixing Model (LMM) is assumed to perform so-called linear unmixing of the data on the basis of a collection of endmembers [**?**, **?**, **?**, **?**, **?**] (akin to the GLM regressors and ICA independent sources).

The LMM can be expressed as follows:

$$\mathbf{x} = \sum_{i=1}^{M} a_i \mathbf{e}_i + \mathbf{w} = \mathbf{E}\mathbf{a} + \mathbf{w},$$

where $\mathbf{x}$ is the d-dimension pattern vector corresponding to the fMRI voxel time series vector, $\mathbf{E}$ is a $d \times M$ matrix whose columns are the d-dimensional vectors, when these vectors are the vertices of a convex region covering the data they are called endmembers $\mathbf{e}_i, i = 1,..,M$, $\mathbf{a}$ is the M-dimension vector of linear mixing coefficients, which correspond to fractional abundances in the convex case, and $\mathbf{w}$ is the d-dimension additive observation noise vector. The linear mixing model is subjected to two constraints on the abundance coefficients when the data points fall into a simplex whose vertices are the endmembers, all abundance coefficients must be non-negative

$$a_i \geq 0, i = 1,..,M$$

and normalized to unity summation

$$\sum_{i=1}^{M} a_i = 1.$$

Under these circumstances, we expect the vectors in $\mathbf{E}$ to be affinely independent and that the convex region defined by them includes *all* the data points. Once the endmembers have been determined, the unmixing process is the computation of the matrix inversion that gives the coordinates of the point relative to the convex region vertices. The simplest approach is the unconstrained least squared error (LSE) estimation given by:

$$\widehat{\mathbf{a}} = \left(\mathbf{E}^T\mathbf{E}\right)^{-1}\mathbf{E}^T\mathbf{x}.$$

Even when the vectors in $\mathbf{E}$ are affinely independent, the coefficients that result from this estimation do not necessarily fulfill the non-negativity and unity normalization. Ensuring both conditions is a complex problem.

We call *Lattice Independent Component Analysis* (LICA) the following approach:

1. Induce from the given data a set of Strongly Lattice Independent (SLI) vectors. These vectors are taken as a set of affine independent vectors, Because of the equivalence between SLI and Affine Independence [**?**, **?**]. The advan-

tages of this approach are (1) that we are not imposing statistical assumptions, (2) that the algorithm is one-pass and very fast because it only uses comparisons and addition, (3) that it is unsupervised and incremental, and (4) that it detects naturally the number of endmembers.

2. Apply the unconstrained least squares estimation to obtain the mixing matrix. The detection results are based on the analysis of the coefficients of this matrix. Therefore, the approach is a combination of linear and lattice computing: a linear component analysis where the components have been discovered by non-linear, lattice theory based, algorithms.

Endmember induction LICA uses some Endmember Induction Algorithm to extract the Lattice Independent Sources (LIS). In this some works we apply the Incremental Endmember Induction Algorithm (IEIA) [**?**, **?**]. In some other works, the Incremental Lattice Source Induction Algorithm (ILSIA). The ILSIA is a greedy incremental algorithm that passes only once over the sample. It starts with a randomly picked input vector and tests each vector in the input dataset to add it to the set of LIS. It is an improved formulation of the Endmember Induction Heuristic Algorithm proposed in [**?**] based on the equivalence between Strong Lattice Independence and Affine Independence [**?**]. There are two conditions for SLI: Lattice Independence and max/min dominance. Lattice Independence is detected based on results on fixed points for Lattice Autoassociative Memories (LAM) [**?**, **?**, **?**], and max/min dominance is tested using algorithms inspired in the ones described in [**?**]. Besides, it uses of Chebyshev best approximation results [**?**] in order to reduce the number of selected vectors.

There are other methods [**?**, **?**, **?**] based on LAAM to obtain a set of SLI vectors. However these methods produce initially a large set of LIS that must be reduced somehow, either resorting to a priori knowledge or to selections based on Mutual Information or other similarity measures. We think that the way the ILSIA the candidate SLI vectors are discarded on the basis of the best approximation in the Chebyshev distance sense is more natural.

2.5 Incremental Lattice Source Induction Algorithm (ILSIA)

The algorithm described in this section evolves from the Endmember Induction Heuristic Algorithm introduced in [**?**]. It is grounded in the formal results on continuous LAAM reviewed in the previous section. The dataset is denoted by $Y = \left\{\mathbf{y}_j; j = 1,\ldots,N\right\} \in \mathbb{R}^{n\times N}$ and the set of LIS induced from the data at any step of the algorithm is denoted by $X = \left\{\mathbf{x}_j; j = 1,\ldots,K\right\} \in \mathbb{R}^{n\times K}$. The number of LIS K will vary from the initial value $K = 1$ up to the number of LIS found by the algorithm, we will skip indexing the set of LIS with the iteration time counter. The algorithm makes only one pass over the sample as in [**?**]. The auxiliary variables $\mathbf{s}_1, \mathbf{s}_2, \mathbf{d} \in \mathbb{R}^n$ serve to count the times that a row has the maximum and minimum, and the component wise differences of the lattice source and input vectors. The expression $(\mathbf{d} == m_1)$ denotes a vector of 0 and 1, where 1 means that corresponding component of $\mathbf{d}$ is equal to the scalar value m_1.

The algorithm goal is to produce sets of SLI vectors extracted from the input dataset. Assuming the truth of conjecture 8 the resulting sets are Affine Independent, and they define convex polytopes that cover some (most of) the data points in the dataset. To ensure that the resulting set of vectors are SLI, we first ensure that they are Lattice Independent in step 3(a) of Algorithm 2.2 by the application of theorem **??**: each new input vector is applied to the LAAM constructed with the already selected LIS. If the recall is perfect, then it is lattice dependent on the LIS, and can be discarded. If not, then the new input vector is a candidate to be included in the LIS. We test in step 3(c) the min and max dominance of the set of LIS enlarged with the new input vector. We need to test the whole enlarged lattice source set because min and max dominance are not preserved when adding a vector to a set of min/max dominant vectors. Note that to test Lattice Independence we need only to build W_{XX} because the set of fixed points is the same for both kinds of LAAM, i.e. $F(W_{XX}) = F(M_{XX})$. However, we need to test both min and max dominance because SLI needs one of them or both to hold. This part of the algorithm is an adaptation of the procedure proposed in [**?**].

If SLI is the only criteria to include input vectors in the set of LIS, then we end up detecting a large number of LIS, so that there will be little significance of the abundance coefficients because many of the LIS will be closely placed in the data space. This is in fact the main inconvenient of the algorithms proposed in [**?**, **?**] that use the columns of a LAAM constructed from the data as the SLI vector set, after

removing lattice dependent vectors. To reduce the set of LIS selected we apply the results on Chebyshev-best approximation from theorem 14 discarding input vectors that can be well approximated by a fixed point of the LAAM constructed from the current set of LIS. In step 3(b) this approximation of a candidate is tested before testing max/min dominance: if the Chebyshev distance from the best approximation to the input vector is below a given threshold, the input vector is considered a noisy version of a vector which is lattice dependent on the current set of LIS. Note that ILSIA that always produces the vertices of simplexes that lie inside the data cloud, so that enforcing the non-negative and normalization conditions of LMM may be impossible for sample data points lying outside the simplex.

Algorithm 2.2 Incremental Lattice Source Induction Algorithm (ILSIA)

1. Initialize the set of LIS $X = \{\mathbf{x}_1\}$ with a randomly picked vector in the input dataset Y.
2. Construct the LAAM based on the strong lattice independent (SLI) vectors: W_{XX}.
3. For each data vector $\mathbf{y}_j$;j=1,...,N
   (a) if $\mathbf{y}_j = W_{XX} \boxtimes \mathbf{y}_j$ then $\mathbf{y}_j$ is lattice dependent on the set of LIS X, skip further processing.
   (b) if $\varsigma\left(W_{XX} \boxtimes \left(\mu + \mathbf{x}^{\#}\right), \mathbf{y}_j\right) < \theta$, where $\mathbf{x}^{\#} = W_{XX}^{*} \boxtimes \mathbf{y}_j$ and $\mu = \frac{1}{2}\left(\left(W_{XX} \boxtimes \mathbf{x}^{\#}\right) \boxtimes \mathbf{y}_j\right)$, then skip further processing.
   (c) test max/min dominance to ensure SLI, consider the enlarged set of LIS $X' = X \cup \{\mathbf{y}_j\}$
      i. $\mu_1 = \mu_2 = 0$
      ii. for $i = 1, \ldots, K+1$
      iii. $\mathbf{s}_1 = \mathbf{s}_2 = \mathbf{0}$
         A. for $j = 1, \ldots, K+1$ and $j \neq i$
            $\mathbf{d} = x_i - \mathbf{x}_j$; $m_1 = \max(\mathbf{d})$; $m_2 = \min(\mathbf{d})$.
            $\mathbf{s}_1 = \mathbf{s}_1 + (\mathbf{d} == m_1)$, $\mathbf{s}_2 = \mathbf{s}_2 + (\mathbf{d} == m_2)$.
         B. $\mu_1 = \mu_1 + (\max(\mathbf{s}_1) == K)$ or $\mu_2 = \mu_2 + (\max(\mathbf{s}_2) == K)$.
      iv. If $\mu_1 = K+1$ or $\mu_1 = K+1$ then $X' = X \cup \{\mathbf{y}_j\}$ is SLI, go to 2 with the enlarged set of LIS and resume exploration from $j+1$.
4. The final set of LIS is X.

Chapter 3

Dendritic Computing

This Chapter contains several approaches to enhance the performance of the single neuron lattice model with dendritic computation (SNLDC). It also includes work performed under the supervision of prof. Gerhard X. Ritter while staying at the University of Florida, consisting in experiments with a novel associative memory based on Dendritic Computing. The contents of the Chapter are as follows: Section 3.1 gives an introduction to the contents of the chapter. Section 3.2 discusses the shrinking hyperbox approach. Section 3.3 describes the hybridization of the SNLDC with kernel and LICA transformation of the data. Section 3.4 introduced the Bootstrapped Dendritic Computation (BDC). Section 3.5 presents image segmentaion with active learning of BDC. Section 3.6 introduces a novel associative memory based on Dendritic Computation. Finally, section 3.7 gives the conclusions of the Chapter.

3.1 Introduction

Dendritic Computing (DC) [**?**, **?**, **?**, **?**, **?**] was introduced as a simple, fast, efficient biologically inspired method to build up classifiers for binary class problems, which could be extended to multiple classes. Specifically the single neuron lattice model with dendrite computation (SNLDC), has been proved to compute a perfect approximation to any data distribution [**?**, **?**]. However it suffers from overfitting problems. Cross-validation experiments result in very poor performance. We found that SNLDC showed high sensitivity but very low specificity in a 10-fold cross-validation experiment. We atribute this to the fact that the learning algorithm

always tries to guarantee the good classification of the class 1 samples. We have followed three ways to enhance the SNLDC performance:

1. We propose to apply a reduction factor on the size of the hyperboxes created by the SNLDC learning algorithm. The results show a better balance between sensitivity and specificity, increasing the classifier accuracy.

2. We perform some data transformations, aiming to have a better data representation for the application of the SNLDC. Specifically, the best results are obtained with a combination of the kernel aproach and the LICA over the original data.

3. Bootstrapped Dendritic Classifiers (BDC) is an ensemble of weak Dendritic Classifiers trained combining their output by majority voting to obtain improved classification generalization performance. Weak Dendritic Classifiers are trained on bootstrapped samples of the train data setting a limit on the number of dendrites. There is no additional data preprocessing. The BDC is also tested in the framework of active learning for image segmentation in a collaboration with Josu Maiora and Borja Ayerdi.

Finally, we use the Dendritic Computation approach to build a novel associative memory with enhanced robustness against noise.

Applications One target application of our work is the classification of Alzheimer's Disease (AD) patients from brain magnetic resonance imaging (MRI) scans. We have worked over a dataset of MRI features extracted from a subset of the OASIS database, as described in Appendix A. Figure 3.1 shows the pipeline summarizing the processes performed up to the classification with the DC system. Computed performance measures are Accuracy, Sensitivity and Specificity. Accuracy is computed as the ratio of correct classifications. Sensitivity is computed as the ratio of true positives to the total number of positive samples. Specificity is computed as the ratio of the true negatives to the total number of negative samples.

The other application is the segmentation of the thrombus in CTA images of Abdominal Aortic Aneurysm (AAA), also explained in Appendix A. The performanc measure is the true positive ratio of the segmentation of the whole volume from a training set extracted from a single slice. Training is performed in an active learning strategy, to minimize the number of selected labeled samples needed to build the classifier.

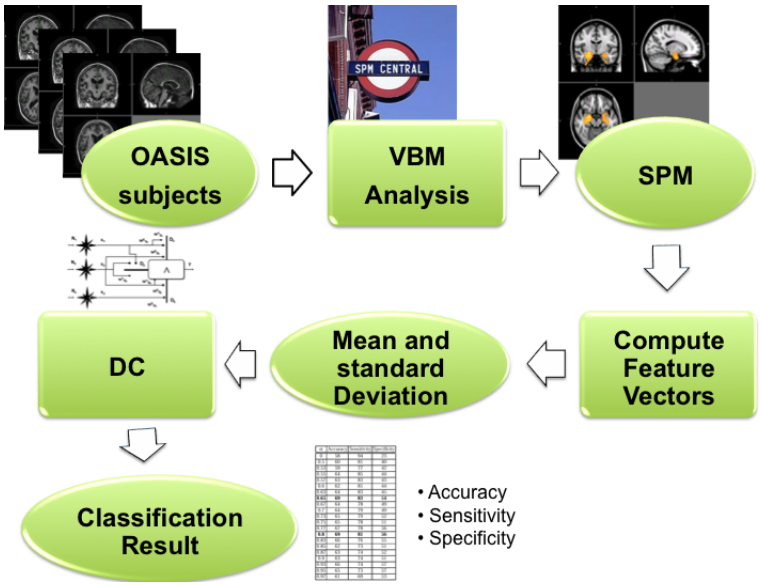


Figure 3.1: Pipeline of the process performed, including VBM, feature extraction and classification by DC

3.2 Optimal Hyperbox shrinking in Dendritic Computing

In this work we propose to modify the basic strategy of hyperbox definition in DC introducing a factor of reduction of these hyperboxes. We obtain a big increase in classification performance applying with this schema over a database of features extracted from Magnetic Resonance Imaging (MRI) including Alzheimer's Disease (AD) patients and control subjects.

3.2.1 DC classification system

We try to produce a better trade-off between the classification specificity and sensitivity by shrinking the boundaries of the box created by the algorithm to exclude the region occupied by a misclassified item of class 0. We define a shrinking factor $\alpha \in [0,1)$ that affects the size of the box created to exclude a region of space from the initial hyperbox that encloses all items of class 1. This shrinking factor is introduced in step 9 of the algorithm 3.1. The effect of this strategy can be appreciated comparing figures 3.2 and 3.3. In figure 3.2 we show the boxes generated by the original learning algorithm. Objects of class 1 correspond to crosses. In

Algorithm 3.1 Dendritic Computing learning algorithm based on elimination with shrinking factor α.

Training set $T = \left\{\left(\mathbf{x}^{\xi}, c_{\xi}\right) \mathbf{x}^{\xi} \in \mathbb{R}^n, c_{\xi} \in \{0,1\}; \xi = 1,\ldots,m\right\}$, $C_1 = \left\{\xi : c_{\xi} = 1\right\}, C_0 = \left\{\xi : c_{\xi} = 0\right\}$

1. Initialize $j = 1$, $I_j = \{1,\ldots n\}$, $P_j = \{1,\ldots,m\}$, $L_{ij} = \{0,1\}$,

$$w_{ij}^1 = -\bigwedge_{c_{\xi}=1} x_i^{\xi}; w_{ij}^0 = -\bigvee_{c_{\xi}=1} x_i^{\xi}, \textit{biologically} \forall i \in I$$

2. Compute response of the current dendrite D_j, with $p_j = (-1)^{\mathrm{sgn}(j-1)}$:

$$\tau_j\left(\mathbf{x}^{\xi}\right) = p_j \bigwedge_{i \in I_j} \bigwedge_{l \in L_{ij}} (-1)^{1-l}\left(x_i^{\xi} + w_{ij}^l\right), \forall \xi \in P_j.$$

3. Compute the total response of the neuron:

$$\tau\left(\mathbf{x}^{\xi}\right) = \bigwedge_{k=1}^{j} \tau_k\left(\mathbf{x}^{\xi}\right); \xi = 1,\ldots,m.$$

4. If $\forall \xi \left(f\left(\tau\left(\mathbf{x}^{\xi}\right)\right) = c_{\xi}\right)$ the algorithm stops here with perfect classification of the training set.

5. Create a new dendrite $j = j+1$, $I_j = I' = X = E = H = \emptyset$, $D = C_1$

6. Select $\mathbf{x}^{\gamma}$ such that $c_{\gamma} = 0$ and $f\left(\tau\left(\mathbf{x}^{\gamma}\right)\right) = 1$.

7. $\mu = \bigwedge_{\xi \neq \gamma}\left\{\bigvee_{i=1}^n \left|x_i^{\gamma} - x_i^{\xi}\right| : \xi \in D\right\}$.

8. $I' = \left\{i : \left|x_i^{\gamma} - x_i^{\xi}\right| = \mu, \xi \in D\right\}; X = \left\{\left(i, x_i^{\xi}\right) : \left|x_i^{\gamma} - x_i^{\xi}\right| = \mu, \xi \in D\right\}$.

9. $\forall \left(i, x_i^{\xi}\right) \in X$

   (a) if $x_i^{\gamma} > x_i^{\xi}$ then $w_{ij}^1 = -(x_i^{\xi} + \alpha \cdot \mu)$, $E_{ij} = \{1\}$

   (b) if $x_i^{\gamma} < x_i^{\xi}$ then $w_{ij}^0 = -(x_i^{\xi} - \alpha \cdot \mu)$, $H_{ij} = \{0\}$

10. $I_j = I_j \bigcup I'$; $L_{ij} = E_{ij} \bigcup H_{ij}$

11. $D' = \left\{\xi \in D : \forall i \in I_j, -w_{ij}^1 < x_i^{\xi} < -w_{ij}^0\right\}$. If $D' = \emptyset$ then goto step 2, else $D = D'$ goto step 7.

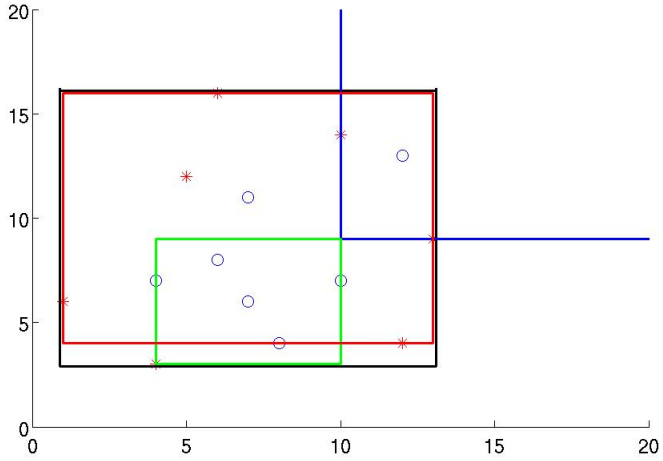


Figure 3.2: Resulting boxes of the original DC learning on a synthetic 2D dataset

figure 3.2 we show the boxes generated by the learning algorithm with shrinking factor $\alpha = 0.8$. It can be appreciated the shrinking algorithm creates more boxes bounding more closely the class 0 items allowing for better generalization of the class 1 results.

3.2.2 Experimental results

For each shrinking parameter value we have performed a 10-fold cross-validation approach, testing more than 50 partitions of the data to obtain each performance estimation. The summary of the best results is presented in Table 3.1 and Figure 3.4 where it can be appreciated that the baseline DC has a poor specificity and a high sensitivity. DC systematically produces low ratios of false negatives, however it produces a large ratio of false positives. Per construction, it is biased towards the positive class C_1. In fact, the main improvement introduced by the shrinking hyperbox approach is an increase in specificity. The DC based approaches have a much higher sensitivity, but their worse specificity degrades their accuracy performance. Varying the shrinking factor α we obtain varying trade-offs between specificity and sensititivity, decreasing the latter while increasing the former. The best results are obtained with $\alpha = 0.8$. In this case the sensitivity is comparable to the results from previous experiments on the same database [**?**], while the specificity is still below

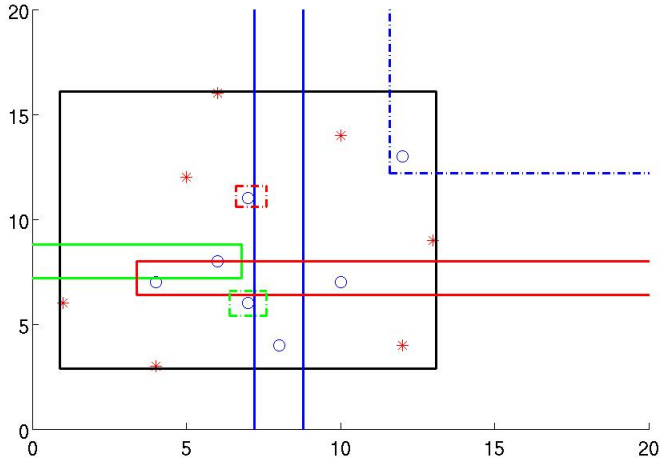


Figure 3.3: Resulting boxes of the DC algorithm with shrinking factor $\alpha = 0.8$.

the results for state of art approaches.

3.3 Hybrid Dendritic Computing with Kernel-LICA

In this Section we explore the use of Lattice Independent Component Analysis (LICA) and the Kernel transformation of the data as an appropriate feature extraction that improves the generalization of SLNDC classifiers. Training is performed by Algorithm 3.1 on the transformed data. First we will review some ideas abotu

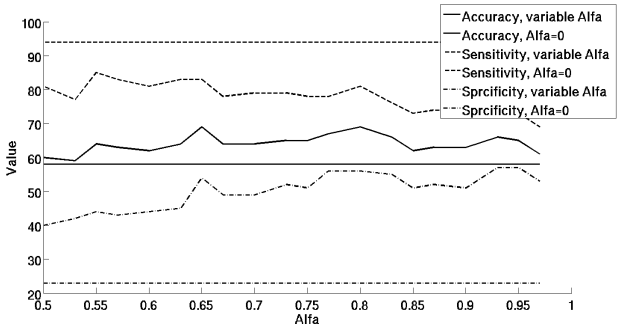


Figure 3.4: DC result varing α and $\alpha = 0$

α	Accuracy	Sensitivity	Specificity
0	58	94	23
0.5	60	81	40
0.53	59	77	42
0.55	64	85	44
0.57	63	83	43
0.6	62	81	44
0.63	64	83	45
0.65	**69**	**83**	**54**
0.67	64	78	49
0.7	64	79	49
0.73	65	79	52
0.75	65	78	51
0.77	67	78	56
0.8	**69**	**81**	**56**
0.83	66	76	55
0.85	62	73	51
0.87	63	74	52
0.9	63	74	51
0.93	66	74	57
0.95	65	73	57
0.97	61	69	53

Table 3.1: Summary of best results of validation experiments over AD MSD feature database. Firs row corresponds to the original DC algorithm[**?**].

Kernel approaches to classification.

3.3.1 Kernel Aproaches.

The kernel transformation has been found very useful in statistics and pattern recognition applications [**?**]. A kernel is a function

$$\kappa(\mathbf{x},\mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle, \tag{3.1}$$

for all $\mathbf{x},\mathbf{z} \in X$, where $X \subseteq \mathbb{R}^n$ is the input pattern space, and ϕ is a mapping into an (inner product) feature space F

$$\phi : X \to F. \tag{3.2}$$

Kernel functions make possible the use of feature spaces with an exponential or even infinite number of dimensions without explicitly computing the features. They are combined with other algorithms as a preprocessing step of the data. In the literature they have allowed to extend linear efficient solutions to non-linear problems. For instance, consider the linear regression problem of finding the linear function $g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle$ that best interpolates a given training set $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}$ with $y_i \in \mathbb{R}$, solved minimizing the function $f(\mathbf{x}, y) = |y - \langle \mathbf{w}, \mathbf{x} \rangle|$ by the well know least squares solution $\mathbf{w} = (\mathbf{X}\mathbf{X}')^{-1}\mathbf{X}'\mathbf{y}$, where $\mathbf{X}$ is the matrix composed of all the sample input vectors, and $\mathbf{y}$ the vector composed of all the labels in the sample. The non-linear extension can be obtained considering a transformation of the sample into the feature space $\hat{S} = \{(\phi(\mathbf{x}_1), y_1), \ldots, (\phi(\mathbf{x}_m), y_m)\}$. The function to be minimized is $f(\mathbf{x}, y) = |y - \langle \mathbf{w}, \phi(\mathbf{x}) \rangle|$. Using a dual approach, the predictive function is reformulated as $g(\mathbf{x}) = \mathbf{y}'(\mathbf{G} - \lambda \mathbf{I})^{-1}\mathbf{k}$, where $\mathbf{G} = \mathbf{X}\mathbf{X}'$ with entries $\mathbf{G}_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, and $\mathbf{k}$ contains the values $k_i = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$. That is, all computations can be performed on the values of the kernel functions, solving the problem with the same procedure employed to solve the linear problem. The kernel matrix $\mathbf{G}$ is the central structure of all the kernel based approaches. For instance, Principal Component Analysis (PCA) of the kernel matrix can be interpreted (with some corrections [**?**]) as a PCA of the data in feature space. Applying the same kind of heuristic, we perform also the LICA over the kernel matrix. The obtained success would indicate the need to examine more closely this approach. Finally,

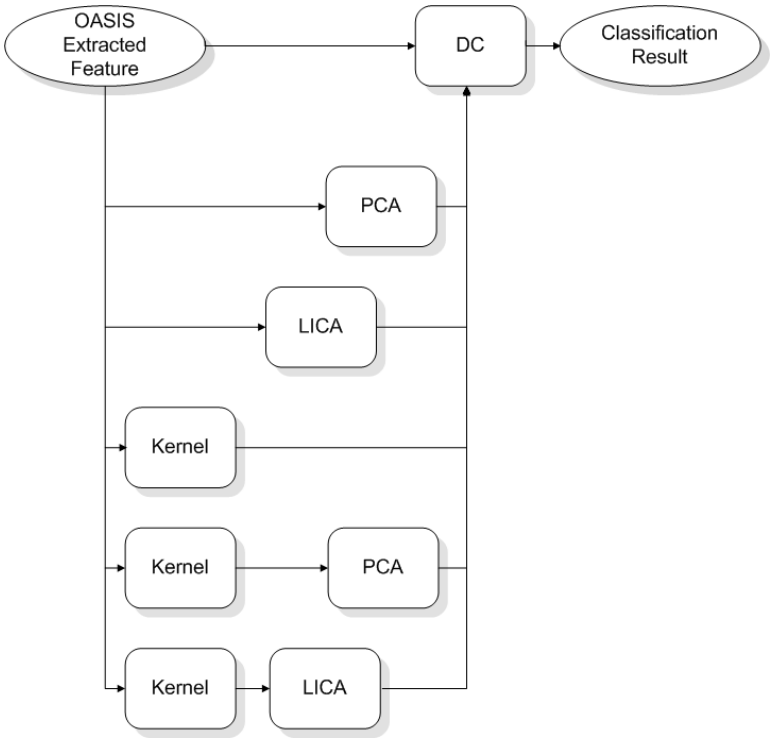


Figure 3.5: The experimental exploration.

we define the Gaussian kernel that will be used in the experiments:

$$\kappa(\mathbf{x},\mathbf{z}) = \exp\left(-\|\mathbf{x}-\mathbf{z}\|^2/2\sigma^2\right). \tag{3.3}$$

3.3.2 Experimental results

Figure 3.1 describes the combinations of systems that we have tested over the AD versus controls database of feature vectors. Each of the possible paths in the graph from the OASIS data up to the classification results corresponds to a combination of systems tested. For each combination we have explored the corresponding parameters in a systematic way, using a 10-fold cross-validation approach, repeated more than 50 times to obtain each performance estimation value in the figures and tables. The 10-fold crossvalidation creates a random partition into ten equally sized subsets of the sample, then each subset in turn is considered as the test data while the remaining data samples are used to build the classifier. Performance measures

are computed as the average of the performances obtained in all repetitions of the training and testing process. Of course, the given values always refer to the test sample data partition, not to the training.

We tested the application of Principal Component Analysis (PCA) to the dimensional reduction of the data previous to DC, the application of LICA to the same end, the transformation of the data with a Gaussian kernel previous to DC or to the application of PCA or LICA. The lower path in figure 3.5 corresponds to the kernel-LICA approach. In the experiments we explored the effect of the diverse parameters. For PCA we computed transformations with up 10 eigenvectors, accounting for 99% of the accumulated eigenvalues. For LICA we tested values in the ranges $\alpha \in [0.01, 0.09] \bigcup [0.1, 0.9] \bigcup [1, 10]$ with corresponding uniform sampling in these intervals. The Gaussian kernel parameter was computed as $\sigma = 10^k$ with $k = [-3, 1]$ sampled uniformly in this interval.

In figure 3.6 we plot the result of PCA-DC as a function of the number of eigenvectors. The average accuracy best result is obtained with one eigenvector and decreases dramatically after that. Figure 3.7 shows the plot of the LICA-DC results as a function of the α parameter that determines the number of endmembers. The best results are for the higher values, which imply less endmembers. Figure 3.8 shows the plot of the DC average accuracy when applied to the Gaussian Kernel transformation of the data with varying σ parameter. The kernel trick seems to work against the DC giving systematically poor results, regardless of the value of its σ parameter. The results of the combination of the Gaussian kernel and PCA are shown in figure 3.9 as surface depending on the number of eigenvectors selected and the value of the σ parameter. It can be appreciated the results are highly sensitive to the kernel parameter, low values giving better results. Overall the kernel PCA-DC transformation improves the results of the PCA-DC combination, although the best result is lower for the Kernel PCA-DC than for the PCA-DC. Finally, figure 3.10 shows the results of the combination of the Gaussian kernel preprocessing with the LICA feature extraction for DC. Values improve with low values of σ and moderate α. Both 3D surface responses in figures 3.9 and 3.10 have embedded the flat surface corresponding to the baseline DC result of 58% accuracy. Therefore the observed peaks correspond to parameter combinations where the combination of systems improves the baseline DC.

Figure 3.11 presents a summary plot of the results of all the approaches tested against the value of their respective parameters. The plot shows that some of the approaches do not improve in any case the baseline Dendritic Computing result.

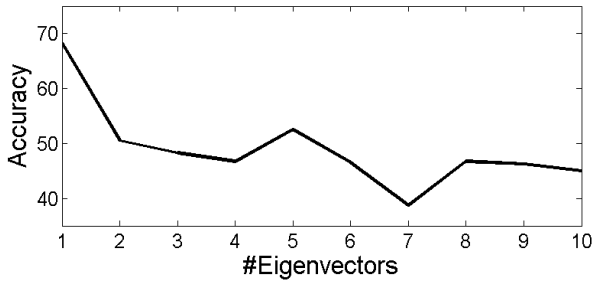


Figure 3.6: PCA-DC results as a function of the number of eigenvectors.

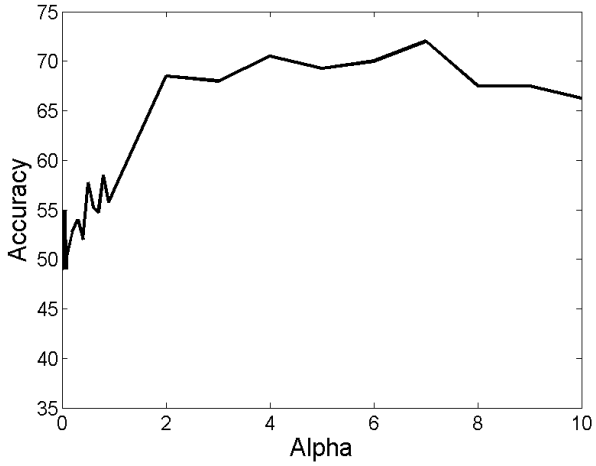


Figure 3.7: LICA-DC results as a function of the noise filter parameter α.

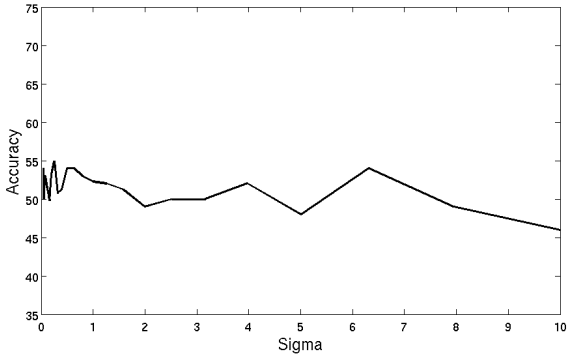


Figure 3.8: DC applied to Gaussian Kernel transformation of the data.

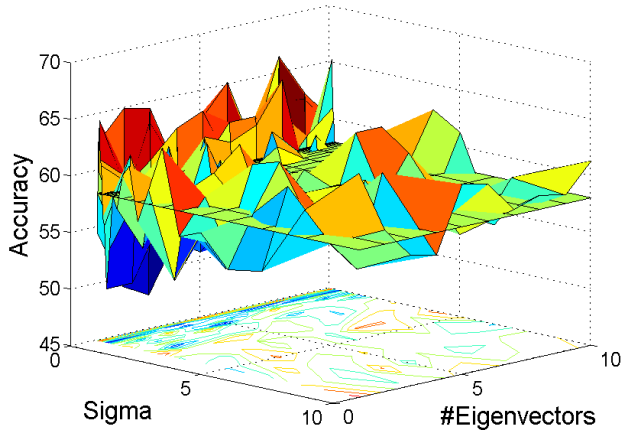


Figure 3.9: Kernel-PCA-DC results varying σ and the number of eigenvectors.

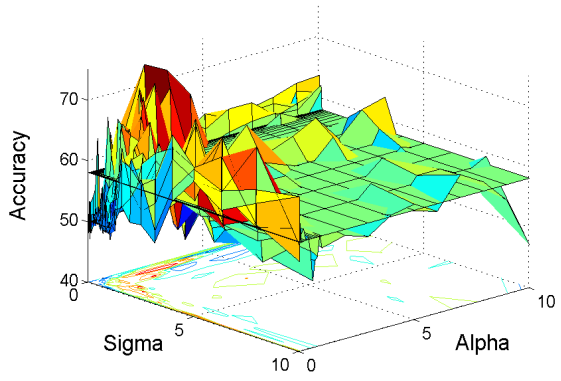


Figure 3.10: Kernel-LICA-DC results varying σ and α.

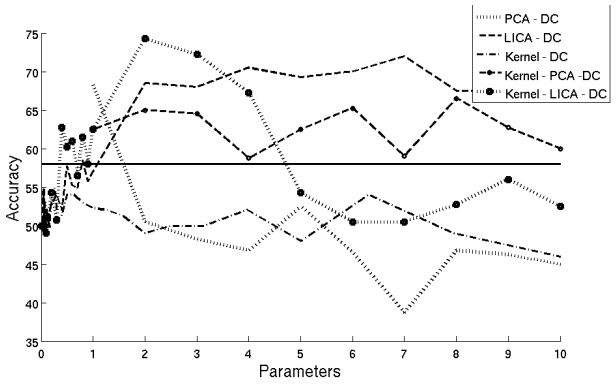


Figure 3.11: Comparative plot of the accuracy of all the approaches tested, the meaning of the parameter axis depends on the approach as illustrated in figures 3.6, 3.7, 3.8, 3.9, and 3.10.

Method	NE	α	σ	Accuracy	Sensitivity	Specificity
DC	-	-	-	58	94	23
PCA - DC	1	-	-	68.25	85.5	51
LICA - DC	1	7	-	**72**	88	56
Kernel - DC	-	-	0.2512	55	98	12
Kernel - PCA - DC	8	-	0.0794	66.5	96	37
Kernel - LICA - DC	3	2	0.5012	**74.25**	96	52.5

Table 3.2: Summary of best results of validation experiments over AD feature database.

The best result is obtained when we apply LICA to a Gaussian kernel transformation of the data. Also we found that the bare application of LICA to the data gives better results than PCA, which only improves DC when reducing the data to one coefficient. The summary of the best results is presented in Table 3.2 where it can be appreciated that the baseline DC has a poor specificity and a high sensitivity. DC systematically produces low ratios of false negatives, however it produces a large ratio of false positives. Per construction, it is biased towards the positive class C_1. In fact, the main improvement introduced by the tested approaches is an increase in sensitivity. Comparing with previous results on this same database [**?**, **?**], we find that the Support Vector Machine (SVM) approach obtains comparable values of sensitivity and specificity. The DC based approaches have a much higher sensitivity, but their worse specificity degrades their accuracy performance.

3.4 Bootstrapped Dendritic Classifiers

The pioneering ensemble classifier was the Random Forest (RF) algorithm proposed in [**?**] that encompasses bagging [**?**] and random decision forests [**?**]. RF became popular due to its simplicity of training and tuning while offering a similar performance to boosting. Bootstrapped Dendritic Classifiers (BDC) is an ensemble of weak Dendritic Classifiers [**?**, **?**, **?**], introduced in Chapter 2, trained combining their output by majority voting to obtain improved classification generalization performance. Weak Dendritic Classifiers are trained on bootstrapped samples of the train data setting a limit on the number of dendrites. There is no additional data preprocessing. This section explores classification performance and the sensitivity to the number of classifiers and the number of dendrites on the classification of Alzheimer's Disease (AD) patients, comparing with previous results obtained on the same feature database obtained from the OASIS database, described in Appendix A.

3.4.1 BDC definition

The Bootstrapped Dendritic Classifiers (BDC) is a collection of DCs,

$$C(x;\psi_j), j = 1,...,N,$$

where ψ_j are independent identically distributed random vectors whose nature depends on their use in the classifier construction. Each DC classifier casts a unit vote for the most popular class of input x. Given a dataset of n samples, a bootstrapped training dataset is used to train DC $C(x;\psi_j)$. The independent identically distributed random vectors ψ_j determine the result of bootstrapping. In conventional RF they also determine the subset of data dimensions $\hat{d}$ such that $\hat{d} \ll d$ on which each tree is grown, here we are not dealing with this kind of DC randomization, which will be studied elsewhere. The main parameters for the experimental evaluation of the BDC are the number of trees and the maximum depth of each dendritic lassifier given by the maximum number of dendrites allowed. Limiting the number of dendrites is a kind of regularizaiton that weakens the classifier. Finally, Algorithm 3.2 specifies the crossvalidation scheme applied in the experiments.

Algorithm 3.2 Crossvalidation scheme for the training of the BDC

Let be $X = \{x_1, \ldots, x_n\}$ input data $x_i \in \mathbb{R}^d$, and $Y = \{y_1, \ldots, y_n\}$ the input data class lables $y_i \in \{0, 1\}$.
N is the number of DC classifiers

1. for i=1:10 (crossvalidation folds)
   (a) select disjoint train $X^e = \left\{x^e_{i_1}, \ldots, x^e_{i_{n-n/10}}\right\} \subset X$, $Y^e = \left\{y^e_{i_1}, \ldots, y^e_{i_{n-n/10}}\right\} \subset Y$ and test $X^t = \left\{x^t_{i_1}, \ldots, x^t_{i_{n/10}}\right\} \subset X$, $Y^t = \left\{y^t_{i_1}, \ldots, y^t_{i_{n/10}}\right\} \subset Y$ datasets .
   (b) For $j = 1 : N$ (construct of classifiers)
      i. Bootstrap a train dataset $X^{eb} = \left\{x^{eb}_{i_1}, \ldots, x^{eb}_{i_{n-2n/10}}\right\} \subset X^e$, $Y^{eb} = \left\{y^{eb}_{i_1}, \ldots, y^{eb}_{i_{n-2n/10}}\right\} \subset Y^e$. Out-of-bag error may be computed on the remaining training data and test $X^e - X^{eb}$, $Y^e - Y^{eb}$, disjunctions.
      ii. Apply DC to train classifier $C_j : \mathbb{R}^d \to \{0, 1\}$ on $\left(X^{eb}, Y^{eb}\right)$.
   (c) end for. Optionally compute out of bag error
   (d) Crossvalidation test, For each $x \in X^t$
      i. compute $C_1(x), \ldots, C_N(x)$
      ii. Majority voting, class $y = 0$ if $\left|\left\{j \left| C_j(x) = 0\right.\right\}\right| > \left|\left\{j \left| C_j(x) = 1\right.\right\}\right|$
   (e) compute accuracy, sensitivity and specificity statistics
2. end fold i

3.4.2 Experimental results

We report the average accuracy, sensitivity and specificity of ten repetitions of 10-fold cross-validation of the BDC developed for AD detection computed over the OASIS dataset explained in Appendix A. The best results found in this computational experiment reported in previous publications for the same MSD features (24 values from each subject MRI volume) are presented in table 3.3[1]. The last rows corresponds to the best results obtained with DC classifiers, while the others correspond to results with other classification algorithms. Specifically, LICA-DC and Kernel-LICA-DC refer to the application of DC after the application of some data preprocessing. In general, DC classifiers have a very low specificity because they target the modeling of the positive class of AD patients. The last row contains the best result found with the proposed BDC which are competitive with the best results found so far. These results have been found in an exhaustive exploration of the effect of the two main parameters, the number of classifiers in the ensemble and the maximum allowed number of dendrites. Figure 3.12 shows the response surface of the Accuracy. It seems that the number of DC classifiers in the ensemble has little effect, though the best average crossvalidation accuracy (89%) was found with large number of classifiers (64). The maximum number of classifiers appears to have some effect: when classifier's number is small - accuracy is still small. Allowing more than five dendrites results in average crossvalidation accuracies above or near 80%. Figure 3.13 shows the response surface of the ensemble sensitivity. The effect of the maximum number of dendrites is very strong, small number of dendrites prevents the overfitting of the classifiers, resulting in low sensitivity because DC tries to model the positive class. As the number of dendrites grows, the BDC ensemble easily reaches very high (close to 100%) sensitivities. Finally, figure 3.14 shows the specificity response surface of the BDC. Specificity remains very low for almost all combinations of number of classifiers and dendrites, and it is the main cause of low classifier performance.

3.5 Active Learning with BDC for image segmentation

We perform the segmentation of medical images following an Active Learning approach that allows quick interactive segmentation minimizing the requirements

[1]Notice that we are labeling as class 0 the AD patients, while in the referred papers [**?**, **?**] they were labeled as class 1.

Classifiers	Accuracy	Sensitivity	Specificity
rbf SVM [?]	81	89	75
LVQ1 [?]	81	90	72
LVQ2 [?]	83	92	74
rbf-DAB-SVM [?]	85	92	78
rbfRVM-LVQ1[?]	87	92	73
LICA - DC [?]	72	88	56
Kernel - LICA - DC [?]	74	96	52.5
Bootstrapped DC	89	100	80

Table 3.3: Results over the MSD features computed from the OASIS data for AD detection

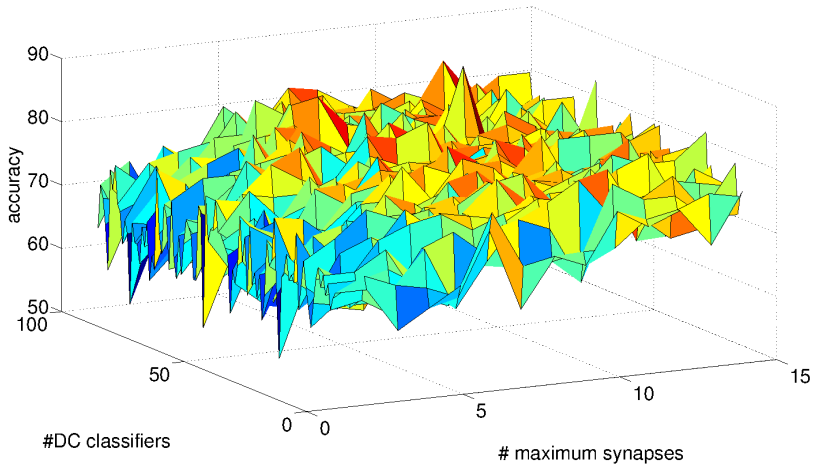


Figure 3.12: Average accuracy for varying number of DC classifiers and maximum number of dendritic synapses

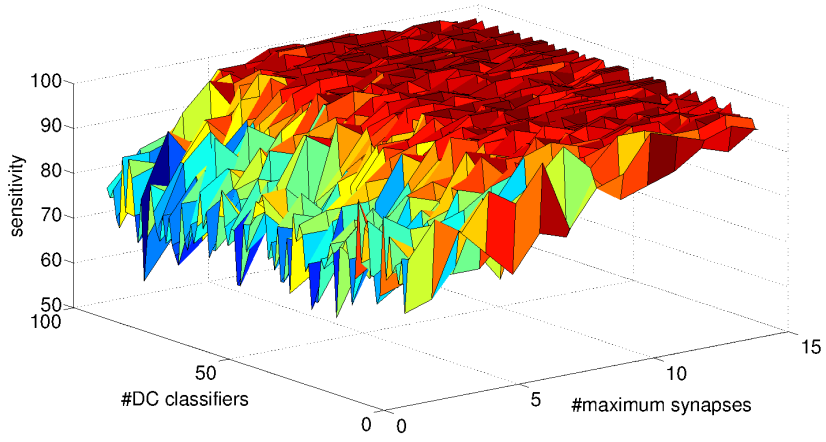


Figure 3.13: Average sensitivity for varying number of DC classifiers and maximum number of dendritic synapses

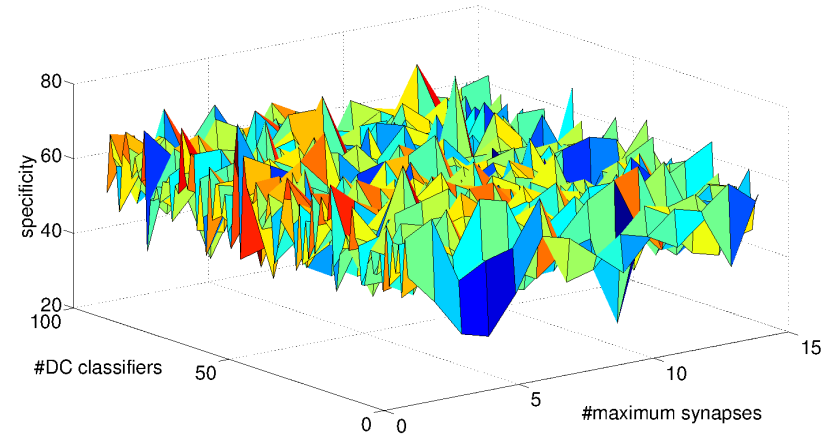


Figure 3.14: Average specificity for varying number of DC classifiers and maximum number of dendritic synapses

for intervention of the human operator. The basic classifier is the Bootstrapped Dendritic Classifier (BDC). We validate the approach on the segmentation of the thrombus in 3D Computed Tomography Angiography (CTA) data of Abdominal Aortic Aneurysm (AAA) patients. The provided ground truth simulates the human oracle. The database is described in Appendix A. The generalization results are given in terms of accuracy and true positive ratio of the classification of the entire volume by the classifier trained on one slice confirm that the approach is worth its consideration for clinical practice.

The approach followed in this paper for AAA thrombus segmentation is to build a voxel classifier into two classes: AAA thrombus or background [**?**, **?**, **?**, **?**]. The data labeling process building the ground truth for the training and validation of the supervised classifiers can be costly and error prone, due to fatigue or operator biases. Active Learning [**?**, **?**] tries to achieve the greatest classifier generalization using the smallest possible training set, minimizing the operator interaction needed to label the training samples. Active Learning starts with a minimal training sample, adding new labeled samples in an iterative process. To obtain the greatest increase in classifier performance, the additional samples are selected according to a classification uncertainty measure[2]. Therefore, the operator is asked to provide the label for the data samples with greatest uncertainty, providing maximum discrimination gain. When there is a great variability in the data, such as in the AAA CTA data, the Active Learning provides a powerful tool to build classifiers from scratch or to perform the adaptation of the previously developed data to new data with different statistical properties. Active Learning has been successfully applied to classification of remote sensing images [**?**, **?**, **?**], image retrieval based on semi-supervised Support Vector Machines [**?**], and the selection of a minimal collection of training images [**?**] for the development of combined generative and discriminative models in the segmentation of CT scans [**?**].

In some aspects, this section is a continuation of the work reported in [**?**]. The classification uncertainty measure is the variance of the class predictions by the individual classifiers in the BDC ensemble. We use almost the same set of features for each voxel, consisting in a collection of the maximum, minimum, median and Gaussian weighted average of the 2D neighborhoods of the voxel of increasing radius. The main contribution in this letter relative to [**?**] is the application of BDC, achieving better results. The experimental setup for validation is as follows: We

[2]The classification uncertainty measure does not require actual knowledge of the data sample label, thus no double-dipping is incurred.

apply one BDC classifier trained by Active Learning on the volume's central axial slice to the remaining axial slices of the volume, in order to test the generalization power of the approach. The Active Learning oracle providing the sample labels in the reported experiments is the ground truth given by a manual segmentation.

3.5.1 Learning and feature selection

3.5.1.1 Active Learning fundamentals

The performance of supervised classifiers strongly depend on the information provided by the data used to train the classifier, so that the appropriate selection and labeling of the training set may be a cumbersome task requiring extensive manual inspection and analysis of the data, typically requiring some visualization tool and labeling of each data sample. Besides, noisy samples may interfere the class statistics, which may lead to poor classification performances and/or over-fitting. For these reasons, a training set must be constructed in a smart way, meaning that it must consists of the minimal set of samples allowing to compute correctly the class boundaries, therefore it must contain the most informative data samples. In the machine learning literature this approach is known as Active Learning.

Active Learning [**?**, **?**] focuses on the interaction between the user and the classifier. Let $X = \{\mathbf{x}_i, y_i\}_{i=1}^{l}$ be a training set consisting of labeled samples, with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1,\ldots,N\}$. Let be $U = \{\mathbf{x}_i\}_{i=l+1}^{l+u} \in \mathbb{R}^d$ the *pool of candidates*, with $u \gg l$, corresponding to the set of unlabeled voxels to be classified. In a given iteration t, the Active Learning algorithm selects from the pool U^t the q candidates that will, at the same time, maximize the gain in performance and reduce the uncertainty of the classification model when added to the current training set X^t. The selected samples $S^t = \{\mathbf{x}_m\}_{m=1}^{q} \subset U$ are labeled with labels $\{y_m\}_{m=1}^{q}$ by an oracle, which can be a human operator in interactive segmentation, or the available ground truth when performing cross-validation experiments. Finally, the set S^t is added to the current training set ($X^{t+1} = X^t \cup S^t$) and removed from the pool of candidates ($U^{t+1} = U^t \backslash S^t$). The process is iterated until a stopping criterion is met, such as the achieved accuracy reaching a preset threshold θ_{max}. Accuracy is computed on the test set, which in our application corresponds to the remaining pixels in the slide. Algorithm 3.3 summarizes the Active Learning process. A critical component of the Active Learning method is the definition of the uncertainty measure associate with each sample, that drives the selection of the samples to be labeled and added to training data. For single classifiers sometimes a histogram of the classifier output

is processed to select the range of values corresponding to uncertain classification [**?**]. For ensemble classifiers, measures on the distribution of the output of the individual classifiers can be used as the direct uncertainty measure [**?**] following a committee approach.

3.5.1.2 Classification uncertainty

The BDC classifiers allow to formulate a committee approach for the estimation of unlabeled sample uncertainty [**?**]: assume that we have built a committee of k base classifiers, i.e. a BDC with k DC classifiers. The output of the committee members provide k labels for each candidate sample $\mathbf{x}_i \in U$. The data sample class label is provided by the majority voting. Our heuristic is that the standard deviation $\sigma(\mathbf{x}_i)$ of the class labels is the measure of the classification uncertainty of $\mathbf{x}_i$. Let us consider an ordering of the pool of candidates $U^* = \left\{\mathbf{x}_{j_i}\right\}_{i=l+1}^{l+u}$, where $\sigma(\mathbf{x}_{j_i}) > \sigma\left(\mathbf{x}_{j_{i+1}}\right)$. The *standard deviation query-by-bagging* heuristic selection of samples to be added to the train set is stated as the following selection:

$$S^t = \left\{\mathbf{x}_{j_m}\right\}_{m=1}^{q} \tag{3.4}$$

Standard deviation of predicted class labels is a natural multi-class heuristic measure of classification uncertainty. A candidate sample for which all the classifiers in the committee agree has a zero prediction standard deviation, thus its inclusion in the training set does not bring additional information. On the contrary, a candidate with maximum disagreement between the classifiers results in maximum standard deviation, and its inclusion will be highly beneficial.

3.5.1.3 Active Learning and feature selection for Image Segmentation

The goal is to classify image voxels into two classes, the target region and the background [**?**]. The Active Learning system returns to the user the unlabeled voxels whose classification outcome is most uncertain with the current classifier. After manual labeling by the user, voxels are included into the training set and the classifier is trained again [**?**]. The feature vector associated with each pixel for its classification is computed using information from its neighboring voxels, applying linear and/or non-linear filtering. In this paper the features initially associated with CTA voxels are: the Euclidean distance of its coordinates to the center of the ground truth in the data domain grid, the voxel intensity, the mean, vari-

Algorithm 3.3 Active learning general algorithm

Inputs

- Initial training set $X^t = \{x_i, y_i\}_{i=1}^{l}$.
- Pool of candidates $U^t = \{x_i\}_{i=l+1}^{l+u}$.
- Number of voxels q to add at each iteration (defining the batch of selected voxels S).

repeat

- Train a classifier with current training set X^t
- **for** *each* candidate in Ut **do**
  - Evaluate classification uncertainty
- **end for**
- Rank the candidates in U^t according to the score classification uncertainty
- Select the q most interesting voxels $S^t = \{x_k\}_{k=1}^{q}$
- The system assigns a label to the selected voxels $S^t = \{x_k, y_k\}_{k=1}$
- Add the batch to the training set $X^{t+1} = X^t \cup S^t$
- Remove the batch from the pool of candidates $U^{t+1} = U^t \setminus S^t$
- Compute accuracy over all test data
- $t = t + 1$

until $accuracy > \theta_{max}$

Table 3.4: Feature importance ranking for the first 10 features selected, specifying the operator used (O), neighborhood radius (R) and the variable importance (VI). Max, Med, GA correspond to Maximum, Median and Gaussian weighted average, respectively

	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
O	Max	Max	Max	Med	Max	Max	Med	Med	Med	GA
R	16	4	8	8	2	1	1	4	16	4
VI	1.277	0.953	0.9531	0.803	0.762	0.759	0.741	0.740	0.732	0.725

ance, maximum and minimum of the voxel neighborhood, for different values of the neighborhood radius (1,2,4...2^n). The definition of these features increases the data dimensionality and the complexity of the classifiers built on them. A first step towards the practical feasibility of the approach (meaning affordable computation times) is the selection of the most informative features, reducing data dimensionality. We follow the results of the feature selection reported in [**?**] on the basis of the variable importance which is defined on the basis of the sensitivity of the predictive generalization on the out-of-box dataset to the perturbations on the value of the feature. To compute the variable importance, the values of the feature in the out-of-box dataset are subjected to randomization, computing the classification error for each permutation of the values of the feature. The average difference of the classification error in the out-of-box dataset with and without randomization is a measure of the sensitivity of the classifier to this feature. For selection, features are ordered by this measure, and a number of them are selected to cover a given percentile of its distribution, namely 95%. The process reported in [**?**] results in the selection reproduced in Table 3.4.

3.5.2 Experiments

3.5.2.1 Experimental setup

Datasets. We have performed computational experiments on 6 CTA datasets to test the proposed Active Learning based image classification approach. Each dataset consists of real human contrast-enhanced datasets of the abdominal area with 512x512 voxel resolution on each slice. Each dataset consists of 216 to 560 slices and 0.887x0.887x1 mm spatial resolution corresponding to patients with Abdominal Aortic Aneurysm. The dataset collection shows a wide diversity of sizes and locations of the thrombus. Some of them have metal streaking artifacts due to the stent

graft placement. Ground truth segmentations of the thrombus for each dataset, that simulate a human oracle providing the labels for the voxels, were obtained manually by a clinical radiologist.

Parameter tuning. The BDC classifier has two sensitive parameters: the number of the classifiers T and their depth given by the number of dendrites D. We set T to 100 and D to 32, from previous empirical results [**?**]. The stopping threshold for the Active Learning has been set to $\theta_{max} = 0.99$. In our experiments we always reach this degree of accuracy on the training set.

Once we get the optimal parameters and feature set, we have designed the following experiment to test our method in the patient CTA volumes, testing the generalization of a single slice classifier: we build just one RF classifier from the data of the central slice of the aneurysm, and we test its generalization with the remaining slices of the CT volume.

Validation. The performance measure of the experiments are the classification accuracy and the ratio of true positives. The accuracy is ratio of correctly classified voxels, that may be misleading when classes are not balanced, as in our application where the background voxels are much more abundant than the thrombus voxels. The second measure is equivalent to the classifier's sensitivity, giving information on how well the target structure, the AAA thrombus, is identified.

3.5.2.2 Experimental Results

Figure 3.15 shows snapshots of the evolution of the Active Learning on one of the volumes. The interval between each snapshot is five iterations. We show the classification uncertainty of each voxel in the images of the left column, white corresponding to the maximum value. The right column shows the actual segmentation obtained by the classifier trained at the corresponding iteration. It can be appreciated that the uncertainty map evolves towards the boundaries of the thrombus, which are the natural places of maximal uncertainty. The segmentation has some alterations until reaching the final result. The classifier learns, forgets and relearns the target class along the learning and sample generation process.

Figure 3.16 shows the segmentation results in the central axial slice of each of the volumes considered. The left column shows the original CTA slice. This visualization helps to highlight the difficulties of the segmentation process: AAAs are of different sizes, with different placements and surrounded by different spatial layouts of structures due to anatomical differences between subjects. Moreover, in

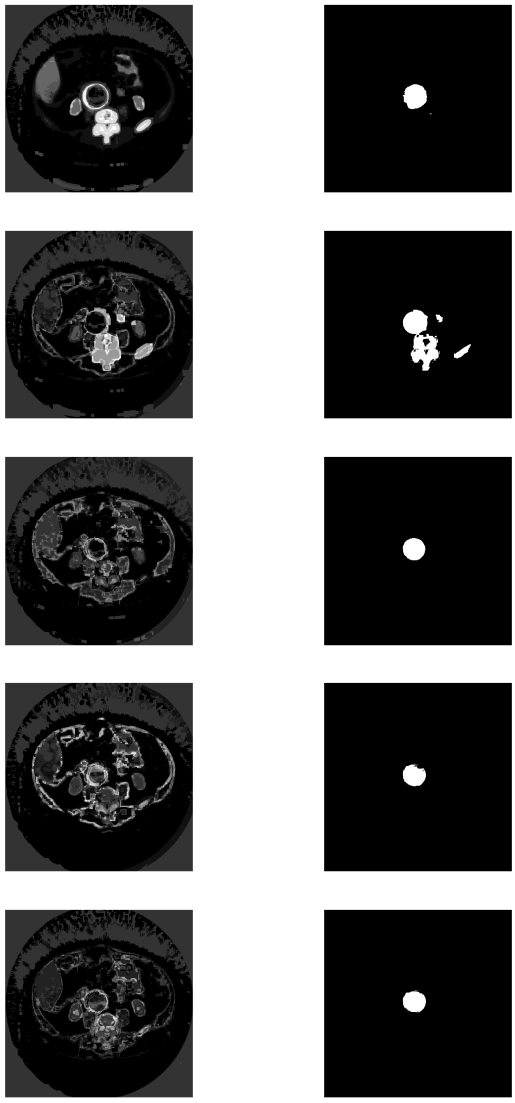


Figure 3.15: Evolution of the active learning process in the central slice of one of the experimental volumes under study, shown at learning iterations 1, 5, 10, 15, and 20. Left column corresponds to the uncertainty value of each voxel. Right column shows the actual thrombus segmentation obtained with the classifier built at this iteration.

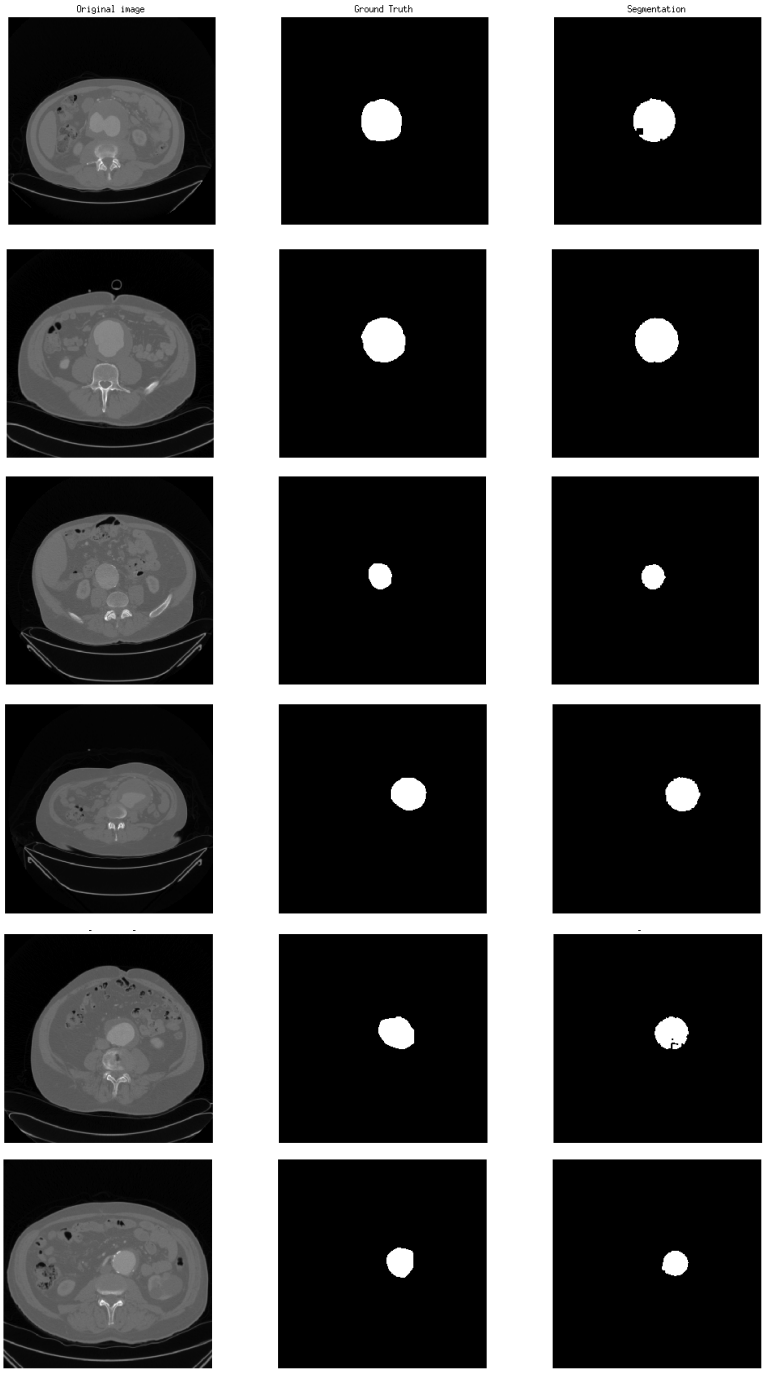


Figure 3.16: Segmentation results in the central slice of the CTA volumes under study after active learning construction of the classifiers. Left column original slice, middle column provided ground truth, right column segmentation achieved by the classifier.

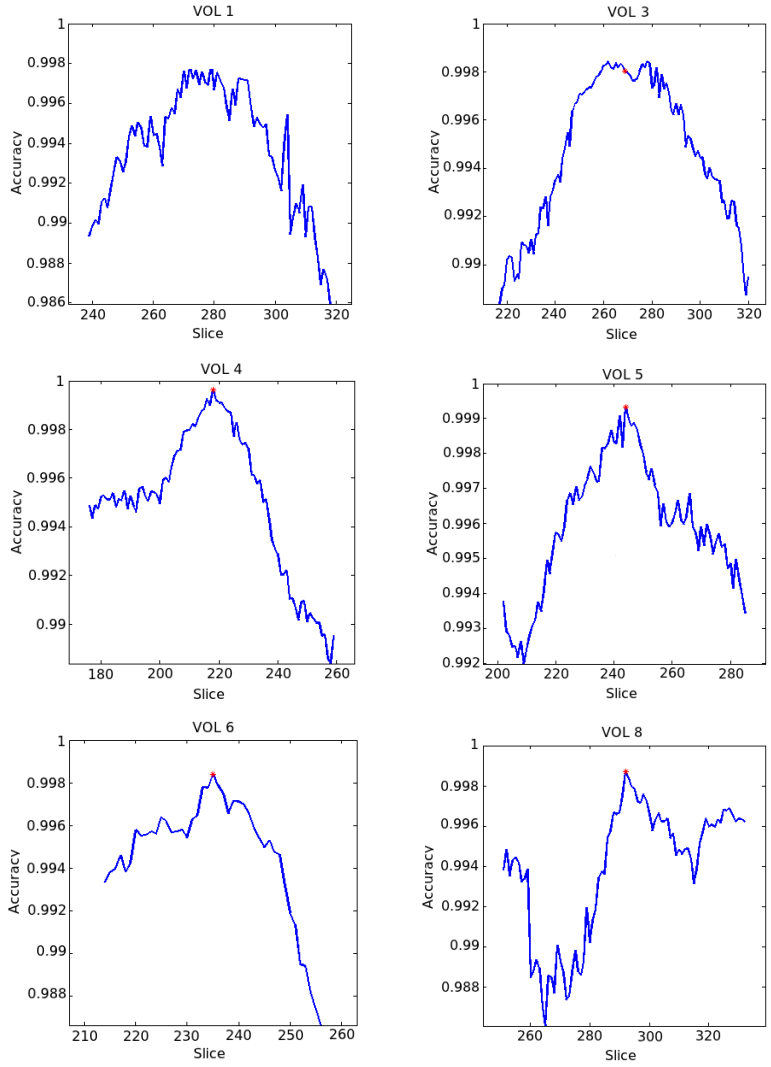


Figure 3.17: Accuracies obtained on the remaining axial slices by the BDC classifier trained on the central axial slice of each of the CTA volumes. Slice numbers are the actual numbers in the volume. The red asterisk identifies the central slice result.

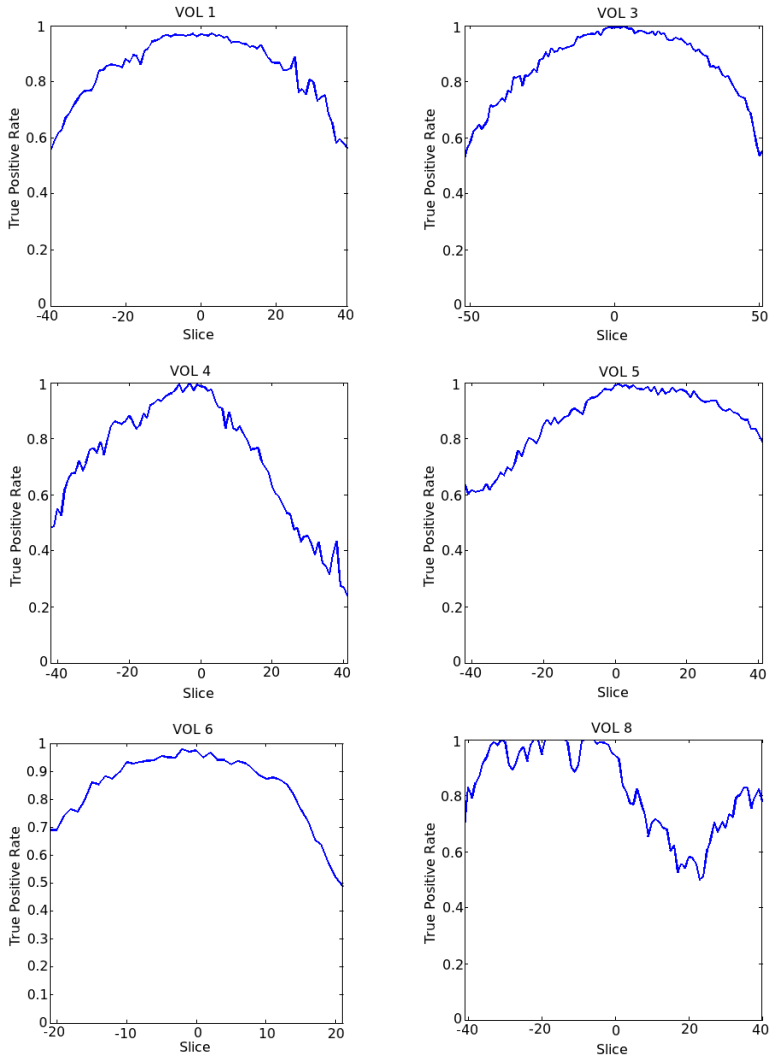


Figure 3.18: True positive rate of the thrombus detection on the 6 CTA volumes when applying the BDC learnt on the central axial slice to the remaining axial slices. Slices are numbered relative to the central slice, positive below it, negative above it.

some cases the actual lumen situated in the middle of the thrombus is hiperintense, but not always. The middle column shows the ground truth segmentation provide by the expert human delineation. The right column shows the segmentation obtained after the Active Learning process in this slice.

Notice that if the BDC classifier obtained on one slice can be applied to the remaining slices without loss of accuracy, the human operator would only need to perform once the Active Learning process to obtain the whole volume segmentation. Fig. 3.17 shows the plots of the accuracy obtained at each CTA volume slice applying the BDC classifier trained on the thrombus central slice for each of the 6 CTA volumes treated in the experiment. The abscissa values correspond to the actual slice numbers in the volume. Obviously, slices where there is no thrombus detected by the ground truth are not included. There is some variability of the plots' span, due to the different sizes of the thrombus in each patient. As can be expected, the drop in classification accuracy is mostly symmetric, but not completely so. The generalization results are very good: the worst accuracy is above 0.98 in almost all cases.

Figure 3.18 plots the true positive ratios obtained at each CTA volume slice applying the BDC classifier trained on the thrombus central slice for each of the 6 CTA volumes treated in the experiment. In those plots, the abscissa's zero value corresponds to the central axial slice, the negative abscissa values correspond to slices above the central slice, the positive values correspond to slices below the central slice. The decrease of the values of true positive ratios are symmetric and show that the system is able to maintain a high sensitivity near the train slice, but that this sensitivity decreases greatly at the extremes of the thrombus. It is feasible to propose additional Active Learning processes in slices where the sensitivity has decreased too much. Detection of such slices can be done based on the growing accumulated uncertainty that can be computed along the classification/segmentation of the slices. Finally, we provide the comparison of average accuracy results published in the literature over the same datasets applying the same training and testing methodology in Table 3.5: training is performed on the central slice of the thrombus, test on the remaining slices, and the accuracy reported is the average over all test slices in the volume. It can be appreciated that BDC provides the best global result in most volumes.

Table 3.5: Comparative average accuracy results published in the literature. Classifiers trained with Active Leraning over one central slice and tested over the remaining data. Classifiers tested are: Random Forest (RF) [**?**] and Hybrid ELM Rotation Forest (HERF) [**?**]. Bold values are the maximum for the corresponding dataset.

	RF	HERF	BDC
Vol. 1	0.993	0.992	**0.994**
Vol. 2	0.980	0.981	**0.985**
Vol. 3	0.992	0.993	**0.995**
Vol. 4	0.995	**0.996**	**0.996**
Vol. 5	0.991	**0.995**	**0.995**
Vol. 6	**0.996**	0.993	0.994
Vol. 7	**0.994**	0.993	0.979
Vol. 8	0.990	0.898	**0.994**

3.6 A Novel Lattice Associative Memory Based on Dendritic Computing

We present a novel hetero-associative memory based on dendritic neural computation. The computations in this model are based on lattice group operations. The proposed model does not suffer from the usual storage capacity problem and is extremely robust in the presence of various types of noise and data corruption.

The matrix correlation memories resulting from the work of Steinbuch, Kohonen, Anderson, and Hopfield were the earliest artificial neural network (ANN) examples of associative memories [**?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**, **?**]. Matrix correlation memories based on lattice computations were first introduced in the late 1990s [**?**, **?**, **?**]. These memories had the advantage of unlimited storage capacity and one step convergence. However, they were susceptible to certain types of random noise. The concept of dendritic computing was partially due to trying to eliminate the noise problem encountered in the construction of artificial memories. The other reason was to provide an artificial neural paradigm that is closer related to actual biological neural computation [**?**].

We the focus is on a novel Dendritic Lattice based (hetero) Associative Memory or, simply, DLAM. Recently two new DLAMs have appeared in the literature [**?**] and [**?**]. The former being a generalization of the DLAMs given in [**?**], while the latter had no predecessor within lattice theory. However, the latter model was

presented as an auto-associative memory. Here we show that the model easily generalizes to a hetero-associative memory. Similar to earlier lattice based associative memories, this new DLAM has unlimited storage capacity in that it can memorize any finite number of association and provides perfect recall for non-noisy input. However, as we shall demonstrate, its greatest advantage over prior associative memories is that it can recall association even when the input is an exemplar pattern that has been corrupted by more than 90% of random noise.

3.6.1 Dendritic Lattice Associative Memories

The Dendritic Lattice based Associative Memory or DLAM described in this section can store any desirable number of pattern associations and has perfect recall when presented with an exemplary pattern. Furthermore, it is extremely robust in the presence of noise and can be applied to both Boolean and real number value patterns.

The proposed DLAM consists of four layers of neurons: an input layer, two hidden layers, and an output layer. The number of neurons in each layer is predetermined by the dimensionality of the pattern domains. Explicitely, if $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\} \subset \mathbb{R}^n$ and $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^K\} \subset \mathbb{R}^m$, then the number of neurons in the input layer is n, in the two hidden layers it is K, and the number in the output layer is m. We denote the neurons in the input layer by $N_1, \ldots, N_n$, in the first hidden layer by $A_1, \ldots, A_K$, in the second hidden layer by $B_1, \ldots, B_K$ and in the output layer by $M_1, \ldots, M_m$. We refer to the first and second hidden layer as the *A-layer* and the *B-layer*, respectively. For a given input pattern $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$, the ith neuron N_i will assume as its value the ith coordinate x_i of $\mathbf{x}$ and will propagate this value through its axonal arborization to the dentrites of the hidden layer neurons. The dendritic tree of each hidden neuron A_j has n single branches $d_{j1}, \ldots, d_{jn}$, and each neuron N_i has two axonal fibers terminating on the synaptic sites located on the corresponding branch d_{ji} of the hidden layer neuron A_j as depicted in Figure 2.3.1. Observe that in this formulation the dendritic branch counter $k = i$, making the extra counter k unecessary. The two synaptic weights associated with the two synaptic sites of d_{ji} will be denoted by a_{ij}^{ℓ} and defined by $a_{ij}^{\ell} = -x_i^j$ for $\ell = 0, 1$. The output of each dendritic branch is denoted by $\tau_i^j(\mathbf{x})$. Here we use the formula given by eqn. 2.7 in order to compute this value. Setting $p_{jk} = -1$ and using the

fact that $I(k) = I(i) = \{i\}$, eqn. 2.7 reduces to

$$\begin{aligned} \tau_i^j(\mathbf{x}) &= -\bigwedge_{\ell=0}^{1}(-1)^{1-\ell}\left(x_i + a_{ij}^{\ell}\right) = -[-(x_i - x_i^j) \wedge (x_i - x_i^j)] \\ &= -[-(x_i - x_i^j) \wedge -(x_i^j - x_i)] = (x_i - x_i^j) \vee (x_i^j - x_i). \end{aligned} \tag{3.5}$$

It follows from eqn. **??** that $\tau_i^j(\mathbf{x}) = 0 \Leftrightarrow x_i = x_i^j$ and $\tau_i^j(\mathbf{x}) > 0 \Leftrightarrow x_i \neq x_i^j$. The value $\tau_i^j(\mathbf{x})$ is passed to the cell body of A_j and its state is a function of the combined values received from its dendritic structure. This state is computed using eqn. 2.7 with $p_j = 1$. Specifically, we have

$$\tau_A^j(\mathbf{x}) = \sum_{i=1}^{n} \tau_i^j(\mathbf{x}) = \sum_{i=1}^{n}(x_i - x_i^j) \vee (x_i^j - x_i) = \sum_{i=1}^{n} |x_i - x_i^j|. \tag{3.6}$$

It follows that each neuron A_j in the A-layer computes the L_1-distance between the input pattern $\mathbf{x}$ and the jth exemplar pattern $\mathbf{x}^j$. That is, $\tau_A^j(\mathbf{x}) = d_1(\mathbf{x}, \mathbf{x}^j)$. The activation function for the A-layer neurons is derived from the identity function, namely

$$f_A(z) = \begin{cases} z & \text{if } z \leq T \\ \infty & \text{if } z > T \end{cases}, \tag{3.7}$$

where T is a user defined threshold. We denote the output of A_j by $s_A^j = f_A(\tau_A^j(\mathbf{x}))$ and the collective output of the A-level neurons by s_A.

The output s_A of the A-layer serves as input to the neurons in the B-layer. Here each neuron B_j has two dendrites d_{j1} and d_{j2}. The dendrite d_{j1} has only one synaptic site on which only an axonal fiber of A_j terminates. The synaptic weight of this synapse is given by $b_{jj}^{\ell} = 0$, with $\ell = 0$. The second branch, d_{j2}, receives input from all the remaining neurons of the A-layer; i.e., from $\{A_1, \ldots, A_K\} \setminus \{A_j\}$. The synaptic weight of the synaptic site on d_{j2} for the terminal axonal fiber of neuron A_r, with $r \neq j$, is given by $b_{rj}^{\ell} = 0$, where $\ell = 1$. To compute the values $\tau_k^j(\mathbf{x})$ for the two dendrites of B_j, we use the general formula

$$\tau_k^j(\mathbf{x}) = p_{jk} \bigwedge_{i \in I(k)} \bigwedge_{\ell \in \mathscr{L}(i)} (-1)^{1-\ell}\left(x_i + w_{ijk}^{\ell}\right) \tag{3.8}$$

which is similar to eqn. 2.7. For $k = 1$ and $i = j$ we have $I(1) = \{1\}$ and $\mathscr{L}(j) =$

$\{0\}$. Setting $p_{j1} = 1$ and employing eqn. 3.8 one obtains

$$\tau_1^j(s_A) = \bigwedge_{i \in I(1)} \bigwedge_{\ell \in \mathscr{L}(j)} (-1)^{1-\ell} \left(s_A^j + b_{jj}^\ell \right) = -s_A^j. \tag{3.9}$$

Similarly, for d_{j2} we have $k = 2$, $i = r$, $I(2) = \{1, \ldots, k\} \setminus \{j\}$, and $\mathscr{L}(r) = \{1\}$. Again setting $p_{j2} = 1$, one obtains

$$\tau_2^j(s_A) = \bigwedge_{r \in I(2)} \bigwedge_{\ell \in \mathscr{L}(r)} (-1)^{1-\ell} \left(s_A^r + b_{jr}^\ell \right) = \bigwedge_{r \neq j} s_A^r. \tag{3.10}$$

The values $\tau_1^j(s_A^j)$ and $\tau_2^j(s_A)$ flow into the cell body of B_j and its state is a function of the combined values received from its dendrites:

$$\tau_B^j(s_A) = \sum_{k=1}^{2} \tau_k^j(s_A) = \tau_1^j(s_A) + \tau_2^j(s_A) = \bigwedge_{r \neq j} s_A^r - s_A^j. \tag{3.11}$$

We consider the two possibilities of $\bigwedge_{r \neq j} s_A^r > s_A^j$ and $\bigwedge_{r \neq j} s_A^r \leq s_A^j$. The first possible case implies that $s_A^j \neq \infty$ and, hence, $s_A^j = d_1(\mathbf{x}, \mathbf{x}^j)$. That is, the pattern vector $\mathbf{x}$ is closer to the exemplar pattern $\mathbf{x}^j$ than any of the other exemplar pattern and within the allowable threshold T. The second possibility implies that either there is another exemplar $\mathbf{x}^r$ which is closer (or just as close) to $\mathbf{x}$ as $\mathbf{x}^j$, or that $\mathbf{x}^j$ surpassed the threshold T. In the first case we want the neuron B_j to send that information to the output neurons while in the second case we do not want B_j to fire. In order to achieve this we define the activation function to be the lattice-based hardlimiter

$$f_B(z) = \begin{cases} 0 & \text{if } z > 0 \\ -\infty & \text{if } z \leq 0 \end{cases}. \tag{3.12}$$

Thus, the output of B_j is given by $s_B^j = f_B[\tau_B^j(s_A)]$ and serves as the input to the output layer M. Each output neuron M_i, $i = 1, \ldots, m$, has only a single dendrite d_{i1} receiving excitatory input from all K neurons of the B-layer. The weight associated with the synaptic site on d_{i1} of the terminal axonal fiber of B_j is defined as $w_{ji}^1 = y_i^j$. Here $j = 1, \ldots, K$ and $i = 1, \ldots, m$. Using eqn. 2.7 to compute the output pattern, we note that since each M_i has only one dendrite d_{i1} we have $k = 1$ (for each i) and $I(1) = \{1, \ldots, , K\}$. Also, since we are dealing with excitatory synaptic responses only, we have that for each $j \in I(1)$, $\mathfrak{L}(j) = \{1\}$. By setting $p_{i1} = 1$, eqn. 2.7 now

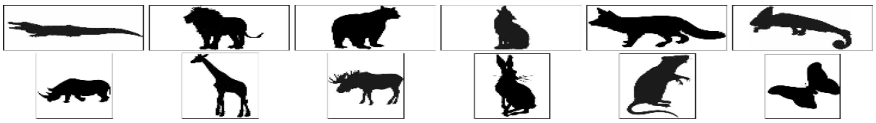


Figure 3.19: Set of Boolean images of six predators in the first row and corresponding six preys in the second row.

reduces to

$$\tau_1^i(s_B) = \bigvee_{j=1}^{K} (s_B^j + w_{ji}^1) = \bigvee_{j=1}^{K} (s_B^j + y_i^j). \tag{3.13}$$

Observe that $\tau^i(s_B) = \tau_1^i(s_B)$ The activation function for each neuron M_i is simply the identity function so that the output y_i of M_i is given by $y_i = \tau^i(s_B)$. The total output of the the set $M_1, \ldots, M_m$ is the vector $\mathbf{y} = (y_1, \ldots, y_m)$. It remains an easy excercise to show that for an uncorrupted input $\mathbf{x}^j$ the output at the M-level will be $\mathbf{y}^j$.

3.6.2 Experiments with Noisy and Corrupted Inputs

In this section we present results of some computational experiments that demonstrate the performance of the proposed DLAM in recalling stored associations when presented with corrupted versions of exemplar patterns. We use images to form pattern vectors only to provide a visual interpretation of the recall. In general. Associative memories are used for pattern recall, not image recall. The transformation of images into vectors is accomplished via the usual column-scan method. We created a database of image patterns from image obtained from various websites.

Experiment 1.

In this experiment, each of the sets X and Y consists of six Boolean exemplar patterns. The set X is derived from the set of six 700×350 Boolean images shown in the top row of Figure 3.19, while the set of associated output patterns is derived from the six 380×500 Boolean images shown in the botton row of Figure 3.19. Thus, $X = \{\mathbf{x}^1, ..., \mathbf{x}^6\}$, with $x^j \in \{0,1\}^{245000}$, and $Y = \{\mathbf{y}^1, ..., \mathbf{y}^6\}$ with $\mathbf{y}^j \in \{0,1\}^{190000}$.

Every pattern image was corrupted adding "salt and pepper" noise. Each noisy pixel of corrupted image is rounded to either 0 or 1 to preserve the Boolean character of the images.

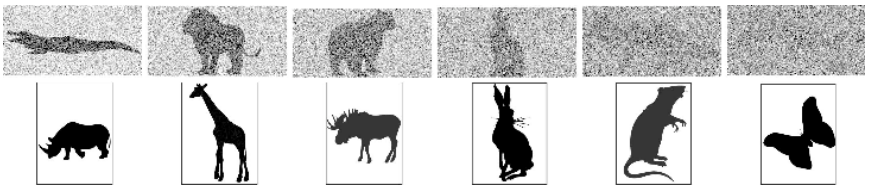


Figure 3.20: First row: Boolean exemplar images corrupted with increasing levels of "salt and pepper" noise of 50%, 60%, 70%, 80%, 90%, and 94% (left to right). Bottom row: Perfect recall associations derived from the noisy input patterns in the top row.

The range of the noise levels varied from 1% to 99% and was tested on all the images. Instances of corrupted input images are shown in Figure 3.20. The corresponding output images recalled by the DLAM are shown in the bottom row. The DLAM shows perfect recall robustness to salt and pepper noise.

Experiment 2

In this example we use a database of grayscale images in which the value of each pixel has an integer intensity value in a range from 0 (black) to 255 (white). Similar to Example 1, we use predator-prey association images as shown in Figure 3.21. Both predator and prey images are of size 265×265. In mathematical terminology we have $X = \{\mathbf{x}^1, ..., \mathbf{x}^K\} \subset \mathbb{R}^{70225}$ and $Y = \{\mathbf{y}^1, ..., \mathbf{y}^K\} \subset \mathbb{R}^{70225}$, $K = 5$. In this experiment we use different types of pattern corruption and noise. Specifically, we simulate noise pattern acquisition by increasing and decreasing image contrast, approximating linear camera motion, applying circular averaging filter, employing the morphological transforms of dilation and erosion with different structuring elements, and by using Gaussian and uniform noise. Figure 3.22 shows some of the tested image corruption changes. Different types of noise corruption have been applied to different images. The first column represents a motion blur, the 2nd Gaussian noise, the 3rd the application of a circular averaging filter, the 4th a morphological erosion with a line as structuring elements and the 5th a morphological dilation with elipsoid as structuring elements.

In the above two experiments, the threshold T for the activation function given in eqn. 7 was set to $T = \infty$; i.e, f_A was simply the identity function. With this threshold, the DLAM performance is very impressive in that associations can be recalled even at 99% random noise levels of the input data. However, images

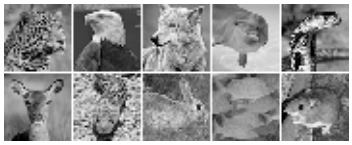


Figure 3.21: Set of grayscale images: 5 Predators in the first row and corresponding 5 Preys in the second row.

Figure 3.22: The exemplar input image patterns are shown in the 1st row. The 2nd through the 4th column below a given predator show the increase in the noise level or image corruption of the predator as discussed in the text. The bottom row illustrates the DLAM's recall performance when presented with a noisy predator image above the prey.

with such high and even lower noise levels of corruption cannot be identified by a human observer when not first shown the original pattern images. This poses the problem of misidentifying intruders. For example, suppose we let $\mathbf{x} \in \mathbb{R}^{70225}$ be obtained from a 265×265 image of a horse and present the DLAM with $\mathbf{x}$ as input. If $T = \infty$, then the DLAM will find the closest L^1-distance to one of the stored images and will associate the horse with one of the predators and correlate it with the predator's prey. To avoid intruders, a threshold $T < \infty$ can usually be determined that avoids misclassification of intruders. In image data (such as shown here) with random noise levels in excess of 60%, most images cannot be recognized by a human observer – the best visual pattern recognizer – when not first shown the corresponding non-noisy exemplar. Thus, if $\bar{x}^j$ represent exemplar x^j corrupted by about 60% of random noise, then setting $T_j = d_1\left(x^j, \bar{x}^j\right)$ and $T = \frac{1}{k}\sum_{j=1}^{k} d_1\left(x^j, \bar{x}^j\right)$ will, generally, present intruders be recognized as a legitimate exemplars. The next example supports this assumption.

Experiment 3

The dataset is the same as in Example 2. The recall of up to 99% of "salt and pepper" noise is perfect just as in Example 1. We consider the response of the DLAM to a new image pattern $\mathbf{x}$ which is not an element of X, namely the horse image of size 265×265 pixels shown in the last column of Figure 3.23.

If we present the image pattern x with the predator image that is closest (in the L^1-distance) to the horse and will, therefore, recall the prey associated with this predator. In this specific case the nearest predator is the leopard as can be ascertained from Table 3.6. Thus, the deer will be associated with the horse when the horse is used as input to the DLAM.

Note that a human observer will have extreme difficulty in identifying any of the images shown in Figure 3.23 if not shown the true exemplars first. Recognition at a noise level of 70% becomes pure guess work. Computing $T_j = d_1\left(x^j, \bar{x}^j\right)$ for each j and each noise level as well as d_1, we can see from Table 3.6 that d_1, where $x^1 = leopard$ and $x = horse$, and $T = \frac{1}{5}\sum_{j=1}^{5} T^j = 5637$ when $\bar{x}^j$ represents as 63% corruption of x^j. Thus, T eliminates x as an intruder. Hence, using $T = 5376$ ($\bar{x}^j$ representing 60% corruption of x^j) would be an even better choice for preventing other intruders.

Noise	0%	50%	60%	63%	65%	70%	80%	90%	100%	Horse
Leopard	0	4470	5374	5634	5813	6297	7158	8066	8932	**5667**
Eagle	0	4492	5348	5626	5844	6252	7154	8080	8947	6293
Wolf	0	4484	5396	5663	5832	6265	7177	8051	8965	6367
Dolphin	0	4452	5385	5640	5816	6281	7162	8059	8952	6713
Cobra	0	4487	5377	5621	5801	6292	7147	8052	8946	6189
Average	0	4477	5376	**5637**	5821	6277	7160	8062	8948	6246

Table 3.6: The distance ($\times 10^3$) between original predator image and the corrupted image with 50%, 60%, 63%, 65%, 70%, 80%, 90% and 100% of "salt and pepper" noise. The last column has the distance to the "horse" image shown in Figure 3.23.

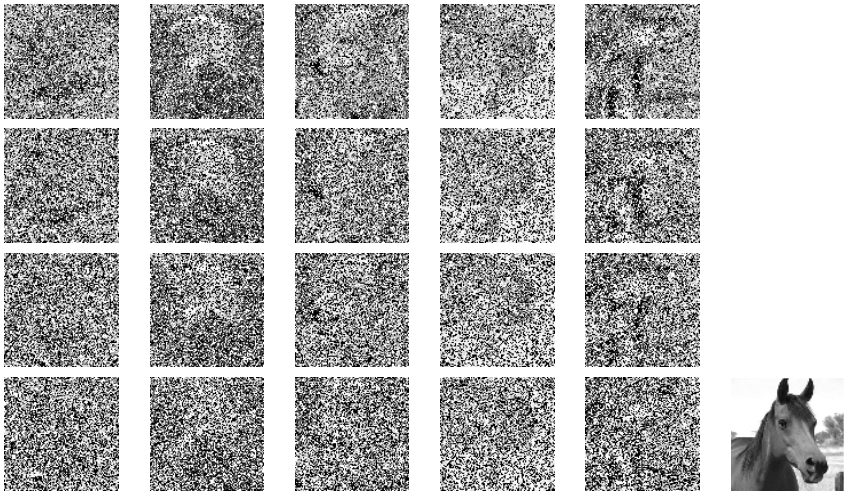


Figure 3.23: Grayscale images from Experiment 3. The 1st, 2nd, and 3rd rows presents the input predator images corrupted with 50%, 60%, 63% "salt and pepper" noise. The 4th row contains corrupted images with the noise parameter set to 70%. These images are at the same distance from the original images as image "horse" in the last column.

3.7 Conclusions

This Chapter addresses several issues turning around the same axis: Dendritic Computing. The general conclusion is that the approach deserves close attention and provides promising results in some instances of the experiments. From the point of view of trying to enhance the performance of the bare SNLDC in cross-validation settings, we have obtained different results:

- The shrinking hyperbox approach provides some balance on the sensitivity and specificifity of the classifiers, improving generalization accuracy. However, the setting of the shrinking parameter is not inmediate.

- The Lattice Independent Component Analysis on a kernel matrix generated applying a Gaussian kernel as the appropriate feature extraction for the Dendritic Computing model. Our approach improves over the application of PCA to the data and to the kernel matrix. Future work can be addressed to develop the theory of the combination of the kernel method with the LICA process.

- The ensemble of DC in a Bootstrapped Dendritic Classifiers (BDC). The main parameters of BDC are the number of DC classifiers and the maximum allowed number of dendrites. We have performed an extensive computational experimentation over a dataset of MRI features for AD patient versus healthy control classification. We have found results which are competitive or improve the best found in the literature for this database. The examination of the response surface shows that the approach seems to be much more sensitive to the number of dendrites than to the number of DC classifiers in the ensemble. Further works will be addressed to more extensive experimentation with diverse conventional datasets.

Besides we have applied the BDC to identify the thrombus in CTA volumes of Abdominal Aortic Aneurysm (AAA) patients. The general approach of Active Learning allows the quick interactive segmentation of the volumes by a human operator performing the labeling of unlabeled voxels with most uncertain classification by the current classifier. BDC allow the definition of a committee based classification uncertainty measure. We present an experimental validation of the approach on a collection of CTA volumes of AAA patients using the provided ground truth to simulate the human oracle. The classifier trained on the axial slice located at the

center of the thrombus is applied to the remaining slices to test the generalization power of the approach. The results show high accuracy across all volume, and very high true positive ratio at slices close to the central one, decreasing towards the spatial limits of the thrombus in the axial direction. This two results imply that there are few spurious detections so that no post-processing of the detected thrombus. To increase the sensitivity of the approach, we propose to reactivate the Active Learning process at slices where the accumulated uncertainty has a big relative growth. This detection can be done in an unsupervised way, amenable for clinical practice where the system will suggest the human operator where more information is required. Further work may be addressed to test on new CTA data provided by local clinicians. Developing an easy to use interface for the radiologist would allow to perform tests in the clinical practice environment introducing minimal disturbances. The extension of this approach to work with 3D volumes instead of 2D slices implies the need to provide appropriate 3D visualization tools to guide the radiologist in 3D space to the selection of the most uncertain voxels.

The final part of the Chapter presents a new hetero-associative lattice memory based on Dendritic Computing. We report experimental results showing that this memory exhibits extreme robustness in the presence of various types of noise. It is our opinion that this DLAM is superior to existing hetero-associative memories. Further work will be addressed to perform exhaustive comparison tests with other associative memory architectures in order to rigorously verify our opinions.

Chapter 4

LICA Applications

This Chapter gives applications of the Lattice Independent Component Analysis (LICA) approach introduced in Chapter 2 to signal unmixing, such as functional Magnetic Resonance Imaging (fMRI) data analysis, incluging task based and resting state data, and Voxel Based Morphometry (VBM) on anatomical MRI data. Some initial works in this Chapter were shared with Maite Garcia Sebastian in her PhD, specifically the analysis of synthetic datasets. Specifically, the contribution of this Thesis to the development of LICA ideas is the realization of some computational experiments which reported in this Chapter.

The contents of the chapter are as follows: Section 4.1 provides an introduction to the LICA and competing methods. Section 4.3 gives results of LICA on synthetic fMRI data. Section 4.4 provides some results on the application of LICA to resting state fMRI for the detection of brain functional networks. Section 4.5 provides the conclusions of this Chapter.

4.1 Introduction

Linear models [**?**] underlie many current data analysis techniques applied to functional Magnetic Resonance Imaging (fMRI) data. The most salient are the General Linear Model (GLM) [**?**, **?**] and the Independent Component Analysis (ICA) [**?**, **?**, **?**]. The GLM is employed in the Statistical Parametric Map (SPM) software package for the analysis of fMRI and other data modalities useful in neurosciences research. In ICA, the data is explained as the linear mixture of statistically independent sources, akin to the GLM regressors. Both the independent sources and

the mixing coefficients are estimated from the data minimizing a suitably defined energy function, which is different among ICA variants.

Lattice Independent Component Analysis (LICA), can be seen as a non-linear relative of Independent Component Analysis (ICA), where ICA statistically independent sources corresponding to LICA Lattice Independent Sources (LIS). LICA uses endmember induction algorithms, such as the Incremental Lattice Source Induction Algorithm (ILSIA), to extract a set of Strong Lattice Independent (SLI) vectors from the input dataset, which define the vertices of a convex polytope covering the input data. Therefore, data linear unmixing on the basis of a given set of Lattice Independent Sources is achieved by least squares error estimation. The linear coefficients correspond to convex coordinates relative to the convex polytope vertices, giving the fractional abundance of the endmembers at each pixel. Hence, LICA is a hybrid of a non-linear lattice based approach for source discovery and a Linear Mixing Model (LMM) of the data. It is unsupervised because the linear model design matrix is induced from the data. Moeover, it does not impose any probabilistic model on the data sources.

Compared with GLM and SPM, ICA and LICA are unsupervised approaches whose regressors are mined from the input dataset. Therefore, the findings of the algorithms are not conditioned by the *a priori* knowledge or expectations. They are less prone to the double dipping [**?**] phenomena arising when the same data is used for selection and selective analysis. They can considered as exploratory data analysis tools and as such they can help discover unsuspected brain connectivity in fMRI data. However to build trust in them, specially in LICA, we need to show that they are able to reproduce the standard results produced by SPM on well known case studies. This is the motivation for the computational experiments reported below.

The main contribution of LICA over the various versions of ICA proposed in the literature is that it does not impose any probabilistic model on the data. Statistical independence is substituted by strong lattice independence which is a property much more easy to find in the data. The computational process of finding the LIS only applies lattice and additive operators, and does not involve the minimization of non-linear energy functions, neither does it need complex data preprocessing, such as whitening and/or dimension reduction. The only kind of normalization required in our works with fMRI data was the shift of the data vectors to the origin subtracting their means.

Applications In this chapter we report applications of LICA to a perform Voxel Based Morphometry (VBM) study on Alzheimer's disease (AD) patients extracted from the OASIS public database (see Appendix). The approach is compared to SPM and Independent Component Analysis results. We show on simulated fMRI data that LICA can discover meaningful sources with efficiency comparable to that of ICA. We explore the network detections obtained with LICA in resting state fMRI data from healthy controls and schizophrenic patients. We compare with the findings of a standard ICA algorithm. We do not find agreement between LICA and ICA. When comparing findings on a control versus a schizophrenic patient, the results from LICA show greater negative correlations than ICA, pointing to a greater potential for discrimination and construction of specific classifiers.

4.2 LICA for VBM

In this section we report results on the realization of Voxel-based Morphometry (VBM) over the OASIS anatomical MRI subset described in Appendix A. Also in this Appendix the description of the conventional VBM process is given. The data for VBM correspond to the patterns of intensities of a voxel across the population of subjects. As implemented in SPM or FSL, the VBM involves the application of GLM to compute the effect of differences between subjects. Here we provide comparative results using ICA and LICA instead of conventional GLM. We have used the FastICA algorithm implementation available at [**?**] . We have also used the implementations of Maximum Likelihood ICA [**?**] (which is equivalent to Infomax ICA), Mean Field ICA [**?**], Molgedey and Schouster ICA based on dynamic decorrelation [**?**], which are available at [**?**].

Figure 4.1 shows the activation results from a FSL study on this data. We have used the preprocessed volumes as inputs for the ICA and LICA algorithms. Detection of significative voxels in ICA and LICA approaches is given by setting the threshold on the mixing/abundance coefficients to the 95% percentil of the empirical distribution (histogram) of this coefficients. We present in figure 4.2 the activation results corresponding to the 3d endmember detected by the LICA algorithm, for comparison with the FSL results. It can be appreciated a great agreement. Because both ICA and LICA are unsupervised in the sense that the pattern searched is not prescribed, they suffer from the identificability problem: we do not know beforehand which of the discovered sources/endmembers correspond to the sought significative pattern, while SPM and FSL approaches are supervised in the sense

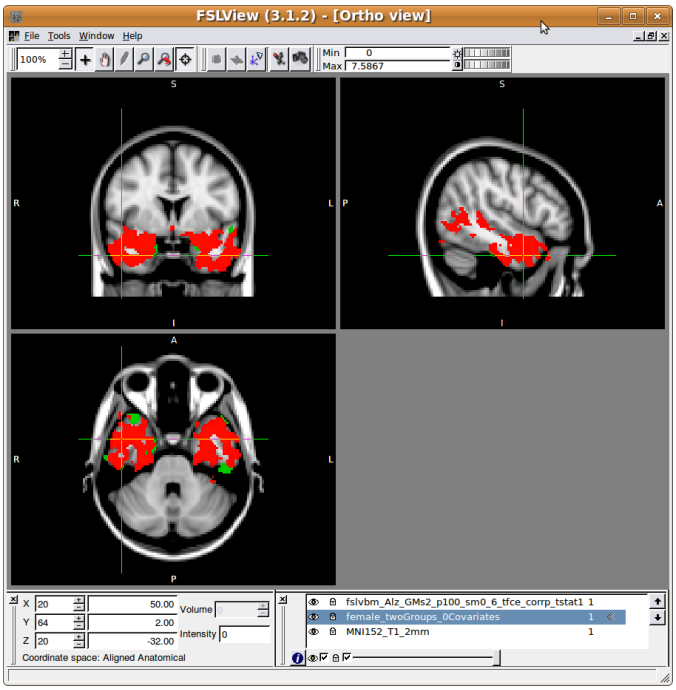


Figure 4.1: FSL significative voxel detection

that we provide the a priori identification of controls and patients, searching for voxels that correlate well with this indicative variable.

In order to provide a quantitative assessment of the agreement between the discoveries of the ICA and LICA and the statistical significances computed by SPM and FSL we computed the correlations between the abundance/mixture matrices of the ICA approach. Table 4.1 shows the correlation between the mixing coefficients and the abundance coefficients of the corresponding ICA ML algorithm sources (the one with best results) and the LICA endmembers, both before (left) and after (right) the application of the 95% percentil threshold to determine the signficative voxels. We decide that the best relation is between the third LICA endmember and the second ICA source, because their correlation does not drop after thresholding, contrary to LICA#4 with ICA#1 whose correlation drops dramatically after thresholding for significance detection.

To give some measure of the meaningfulness of the unsupervised approaches, we must find out if they are able to uncover something that has a good agreement with the findings of either SPM or FSL approaches. Therefore we compute

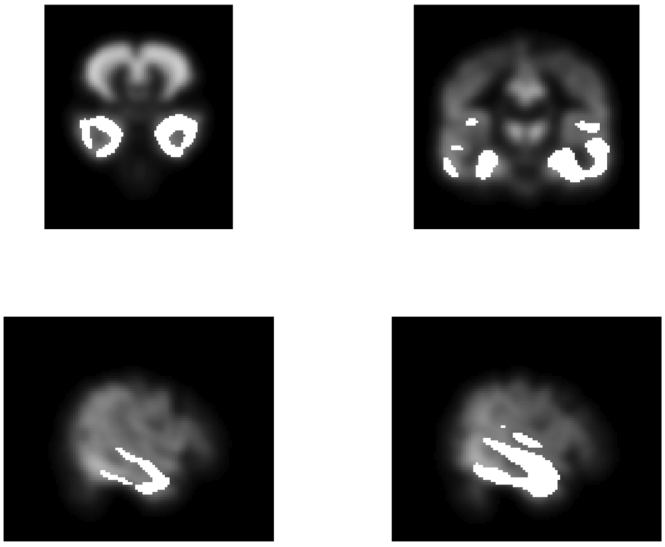


Figure 4.2: LICA activation results for the endmember #3

	ICA ML					ICA ML			
LICA	#1	#2	#3	#4	LICA	#1	#2	#3	#4
#1	0.05	0.24	0.44	-0.01	#1	0.003	0.09	0.34	0.03
#2	0.19	0.12	-0.28	-0.60	#2	0.15	0.05	-0.02	-0.02
#3	0.38	0.67	0.30	0.24	#3	0.01	0.66	0.007	0.08
#4	0.69	0.04	0.26	-0.18	#4	0.26	-0.01	0.13	-0.00

Table 4.1: Correlation among ICA and LICA mixing coefficients, before (left) and after (right) thresholding for activation detection

the correlation between the mixing/abundance coefficients of ICA/LICA and the statistics computed by SPM and FSL. Table 4.2 shows these correlations. Here the agreement between the third endmember of LICA and the secod source of ICA ML obtains a further support, because both are the ones that show maximal agreement with SPM and FSL, and in both ICA and LICA the agreement with FSL is greater than with SPM results.

	#1	#2	#3	#4
ICA vs SPM	-0.11	**0.32**	-0.02	0.02
LICA vs SPM	-0.03	-0.03	**0.23**	-0.06
ICA vs FSL	0.08	**0.56**	0.03	0.07
LICA vs FSL	0.07	0.02	**0.58**	0.20

Table 4.2: Agreement between SPM, FSL, ICA and LICA

4.3 LICA for synthetic fMRI data analysis

We have used the simulated fMRI data [**?**, **?**] [1]. In fMRI the spatial distribution of data sources can be classified into locations of interest and artifacts. The locations of interest include task-related, transiently task-related, and function-related locations. Their spatial distribution are typically super-gaussian in nature because of the high localization of brain functionality. A task-related location and its corresponding source closely match the experimental paradigm. A transiently task-related source, on the other hand, is similar to a task-related source but with an activation that may be pronounced during the beginning of each task cycle and may fade out or change as time progresses. Functional locations are those activated areas which are related to a particular functional area of the brain and the corresponding source may not exhibit a particular pattern. The class of uninteresting sources or artifacts include motion related sources due to head movement, respiration, and cardiac pulsation. Figure 4.3 shows the sources used to produce the simulated fMRI data. Source #1 corresponds to the task related time course, source #6 corresponds to a transient task-related time course. Figure 4.4 shows the spatial distribution of the locations of the sources, corresponding to the mixing matrices in the linear models of both ICA and LICA. Spatial locations #1 and #6 are the ones with most interest from the task point of view. To form the fMRI mixture, first the image data is reshaped into vectors by concatenating columns of the image matrix. The source matrix is multiplied by the time course matrix to obtain a mixture that simulates 100 scans of a single slice of fMRI data.

We have applied the LICA and MS-ICA algorithms to this simulated data. We obtain five LIS with the LICA approach using standard settings of the algorithm, and we set the MS-ICA number of sources to that number. Figure 4.5 presents the LIS found by ILSIA, with the best correlated simulated time course overlaid

[1] Simulated data can be generated with the tools provided in http://mlsp.umbc.edu/simulated_fmri_data.html

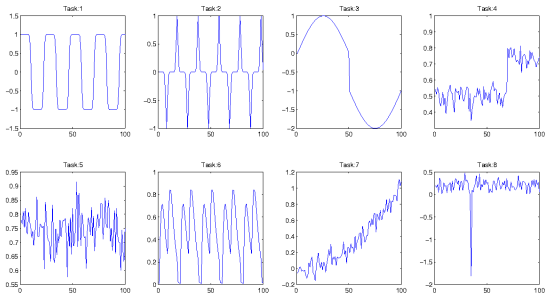


Figure 4.3: Simulated sources (time courses) in the experimental data.

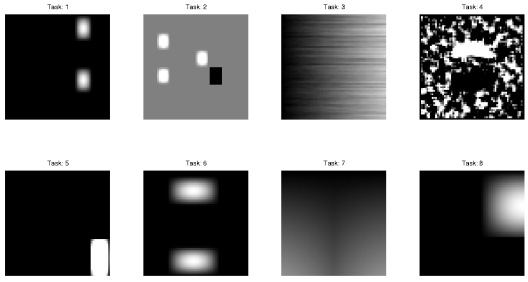


Figure 4.4: Simulated spatial distribution of location of the sources

in red. Figure 4.6 shows the sources found by the MS-ICA together with the best correlated simulated time course. We observe that ILSIA finds both a task related and a function-related source. We show in figure 4.7 the abundance images that correspond to the spatial distributions of the LIS. Notice that the spatial location of the task-related source is well detected in the second image, while the transient task-related source location is also well detected despite that it does not appear as one of the best correlated sources in figure **??**. Because the LICA and ICA algorithms are unsupervised, they can discover sources which indirectly help discover the spatial locations of interest, although the sources themselves are not precise matches of the underlying true sources. Figure 4.8 shows the spatial distribution of the MS-ICA sources. The detection is noisier than that obtained by LICA, and the task-related spatial locations are less clearly detected by MS-ICA. Table 4.3 contains the quantitative measure of the goodness of spatial discovery, given by the Mutual Information similarity measure between the simulated spatial distributions of the simulated sources and the mixing coefficients of MS-ICA and abundances of LICA that give the estimation of the spatial distribution of the discovered sources. We have highlighted the maximum values per column, and we have highlighted the closest one when it is near the maximum of the column. When two columns have their maximum value (or one very close to it) in the same row, it means that the true source is not univocally identified with one of the discovered sources. This is an ambiguous situation that needs to be avoided in some applications. The MS-ICA has more ambiguous columns than the LICA, which is in agreement with the visual assessment of figures 4.7 and 4.8. Sources #4, #5 and #8 are not identified by any of the algorithms. Source #7 is not identified by LICA. Overall the results of LICA are comparable or better, depending on the value given to the discovered sources, than those of MS-ICA.

4.4 LICA detections in resting state fMRI

In this section, we compare the results of networks detected on resting state fMRI (res-fMRI) by ICA and LICA approaches. We have used the FastICA algorithm implementation [**?**]. Because both ICA and LICA are unsupervised in the sense that the pattern searched is not predefined, they suffer from the identifiability problem: we do not know beforehand which of the discovered independent sources/endmembers will correspond to a significant brain connection. Therefore, results need a careful assessment by the medical expert. We will not give here any neurological conclu-

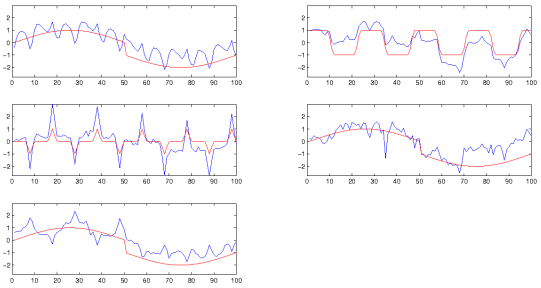


Figure 4.5: Sources found by the ILSIA on the simulated data

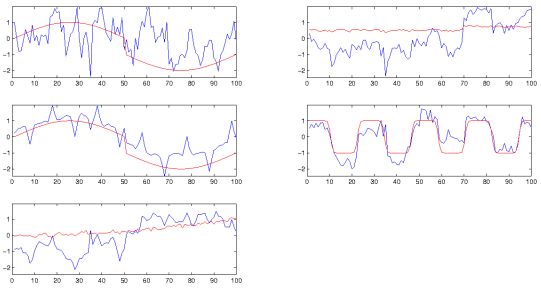


Figure 4.6: Sources found by MS-ICA on the simulated data

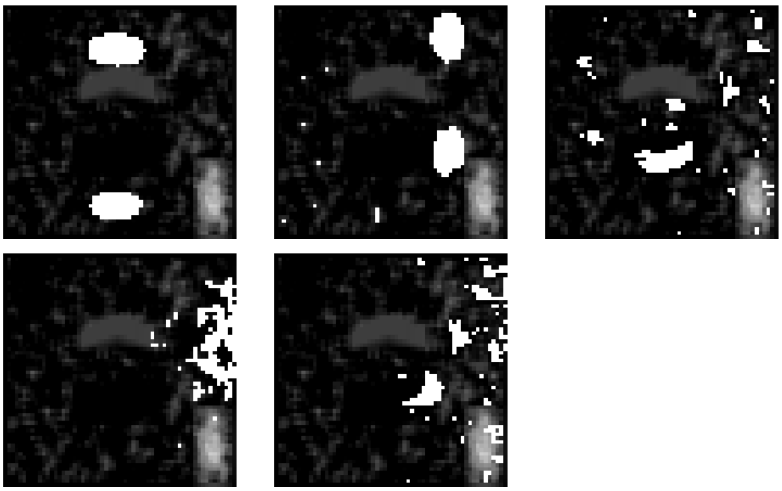


Figure 4.7: Spatial distributions found by LICA on the simulated data.

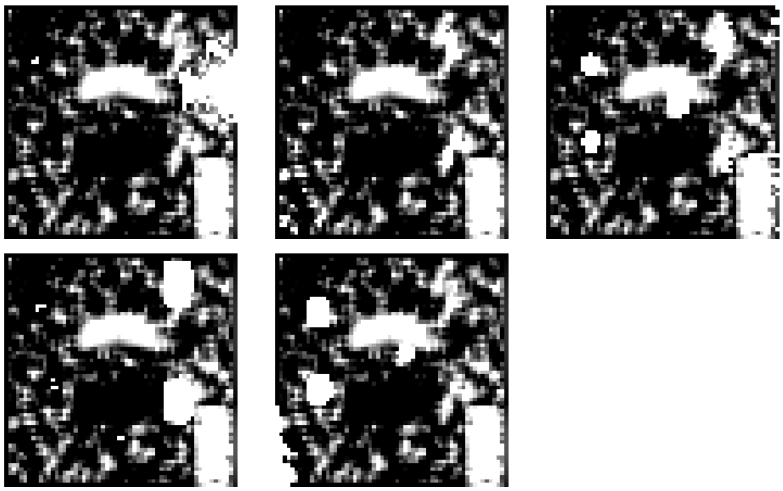


Figure 4.8: Spatial distribution of the sources given by the mixing matrices of MS-ICA on the simulated data.

	MS-ICA					LICA				
Source	#1	#2	#3	#4	#5	#1	#2	#3	#4	#5
#1	-0,16	0,15	0,03	**1,18**	-0,36	-0,52	**2,51**	-0,37	-0,23	0,02
#2	-0,45	-0,78	-0,13	-0,38	-0,29	-0,48	-0,33	**2,26**	-0,13	-0,38
#3	**1,29**	**1,09**	**2,32**	**0,79**	**1,18**	0,31	1,66	0,25	**2,42**	**2,24**
#4	-0,28	0,68	-0,56	-0,63	-0,38	-0,57	-0,72	-0,53	-0,52	-0,50
#5	-1,42	-1,05	-0,80	-0,69	-0,70	-0,71	-0,76	-0,70	-0,57	-0,72
#6	**1,33**	-0,79	-0,30	-0,39	-0,60	**2,26**	-0,44	0,36	-0,50	-0,54
#7	0,62	**1,51**	0,17	-0,24	**0,80**	0,30	-0,20	-0,57	-0,01	0,55
#8	-0,92	-0,81	-0,72	-0,63	-0,62	-0,59	-0,71	-0,68	-0,47	-0,67

Table 4.3: Mutual Information similarity between the spatial locations discovered by LICA and MA-ICA and the ground truth spatial locations.

sion. The results shown in this section are explorations over resting state fMRI data obtained from a healthy control subject and an schizophrenia patient with auditory hallucinations selected from an on-going study in the McLean Hospital. Details of image acquisition and demographic information are given in Appendix A. For each subject we have 240 BOLD volumes and one T1-weighted anatomical image and functional images.

The application of LICA with nominal parameters give 8 endmembers. Accordingly we have computed fastICA setting the number of independent sources to 8. We compute the LICA abundance distributions. For each endmember, we set the 95% percentile of its abundance distribution as the threshold for the detection of the corresponding endmember in the abundance volume. We do the same with the ICA mixture distributions. To explore the agreement between ICA and LICA detections, we have computed the Pearson's correlation between the abundance/mixing volumes of each source/endmember, shown in table 4.4 for the schizophrenia patient and in table 4.5 for the control subject. In both cases, agreement between detections of LICA and ICA is low. The best correlation is ICA #8 versus LICA #5 for the schizophrenia patient. For a visual assessment of the agreement between both detection analysis, we show in figure 4.9 the detections obtained by both algorithms applying the 95% percentile on their respective mixing and abundance coefficients. In this figure, the detection found by LICA is highlighted in blue and the detection found by ICA in red. Overlapping voxels appear in a shade of magenta. It can be appreciated that the LICA detections appear as more compact clusters. Some spurious detections are shown in the sorroundings of the brain due to the diffusion produced by the smoothing filter. From these results is clear that we can not use ICA to validate the findings of LICA.

For further comparison, we have computed the correlations intra-algorithm of the patient versus the control data, meaning that we compute the correlations of the abundance/mixing volumes obtained by the LICA/ICA on the patient and the control data. The aim is to get an idea of the ability of each approach to produce discriminant features. If we find negative correlations of high magnitude then we can say that the corresponding approach has a great potential to generate features that discriminate patients from controls. Table 4.6 shows the correlations between the LICA abundances obtained from the patient and the control subject. Table 4.7 shows the same information for the mixing coefficients of ICA. In these tables we are interested in finding the most negatively correlated detections, implying complementary detections. We have highlighted in bold the negative correlations below

	ICA							
LICA	#1	#2	#3	#4	#5	#6	#7	#8
#1	0.02	0	-0.04	-0.02	0.02	0.03	0.01	0.01
#2	0.03	0.08	-0.1	-0.04	0	0.01	-0.33	0
#3	-0.01	**0.36**	0.01	-0.07	-0.01	-0.02	0.13	-0.01
#4	0.03	0	-0.03	-0.11	0	-0.01	0	-0.01
#5	-0.03	-0.02	**0.16**	-0.01	-0.01	-0.11	-0.02	**0.46**
#6	0.38	-0.03	0.01	-0.13	**0.17**	0	-0.01	0.01
#7	0	-0.02	0.06	-0.02	-0.01	-0.02	-0.02	-0.01
#8	0.25	0.01	-0.22	0.04	-0.52	0.05	0.02	-0.05

Table 4.4: Pearson's Correlation coefficients between ICA and LICA source/endmember detections for the schizophrenia patient.

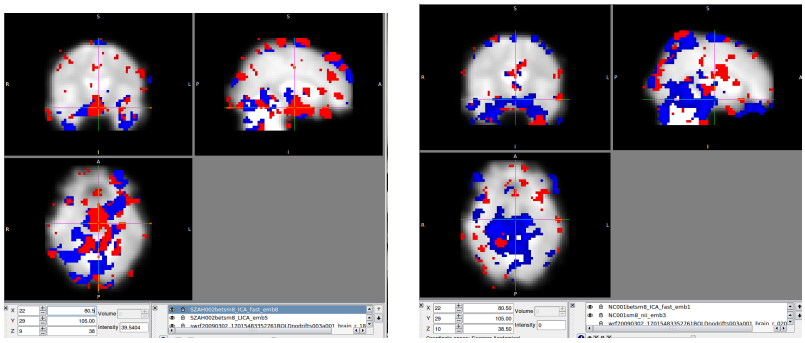
	ICA							
LICA	#1	#2	#3	#4	#5	#6	#7	#8
#1	0.04	-0.02	-0.04	0.05	-0.06	-0.07	0.03	0.01
#2	-0.02	0.02	0	-0.08	-0.03	-0.03	**0.15**	-0.01
#3	**0.22**	-0.05	**0.13**	0.06	0.01	0.08	-0.03	0.07
#4	0.05	-0.22	0.06	-0.09	0.06	0.08	0.08	-0.03
#5	0.03	0.07	-0.07	**0.12**	**0.14**	-0.13	0.04	-0.01
#6	0.04	0	0.05	-0.06	-0.1	0.02	-0.05	-0.03
#7	0.08	**0.1**	0	0.03	-0.03	-0.02	0.09	0.03
#8	-0.02	-0.04	0.02	0.04	-0.05	-0.07	0.07	0.03

Table 4.5: Pearson's Correlation coefficients between ICA and LICA source/endmember detections for the healthy control.

-0.15. We show in figure 4.10 the detections with greatest negative correlation between patient and control for both LICA and ICA. In this figure, red corresponds to the patient volume detection, blue corresponds to the control volume detection. Notice again that LICA detections produce more compact clusters. The greatest discrimination is obtained by LICA

4.5 Conclusions

The LICA approach was introduced in Chapter 2. In this Chapter we have reported results on several applications of LICA to brain medical image, specifically



(a) (b)

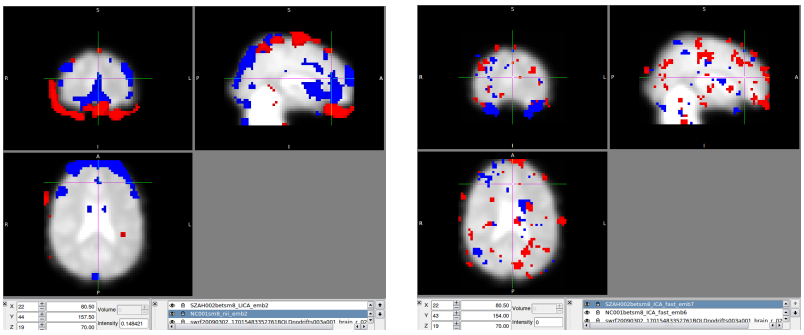
Figure 4.9: Simultaneous visualization of the best correlated detection results from LICA and ICA from tables 4.4 and 4.5 . Red corresponds to ICA detection, Blue to LICA detection. (a) Patient, (b) Control.

	patient							
control	#1	#2	#3	#4	#5	#6	#7	#8
#1	-0.17	-0.04	-0.03	0.24	0.08	0.08	0.09	0.01
#2	0.02	**-0.21**	0.04	0.1	0.15	0.09	0.02	-0.09
#3	**-0.32**	0.05	-0.05	0.14	0.24	0.13	0.13	0.15
#4	0.01	0.15	0.08	0.05	-0.03	0.02	0.05	0.17
#5	-0.14	-0.13	**-0.18**	0.13	0.14	0.11	0.12	-0.04
#6	0.01	-0.06	0.11	0.02	0.02	0.02	-0.02	-0.02
#7	0.06	-0.11	-0.05	-0.15	-0.05	-0.05	-0.12	0.03
#8	**-0.32**	-0.19	-0.02	0.23	0.22	0.2	0.05	0.02

Table 4.6: Correlation between patient and control detections obtained by LICA

	patient							
control	#1	#2	#3	#4	#5	#6	#7	#8
#1	0.41	0.01	-0.15	0.01	**-0.18**	0.02	0.02	-0.04
#2	-0.12	0.02	0.06	-0.04	0.08	-0.01	0.01	0.05
#3	0.02	-0.02	**-0.24**	-0.02	0.01	0.01	0	0.03
#4	0.03	0	0	0	0.02	0.02	0.02	0
#5	0.04	-0.01	0.06	-0.03	-0.05	-0.01	0.01	0.36
#6	0.04	0.07	-0.05	0.01	0	0	**-0.25**	0
#7	0.03	0	-0.02	0	-0.01	0.06	0	-0.03
#8	0.02	0.03	-0.01	0	-0.02	0	0.01	0

Table 4.7: Correlation between patient and control detections obtained by ICA



(a) (b)

Figure 4.10: Findings in the patient versus the control. Greatest negative correlated detections (a) found by LICA, (b) found by ICA

anatomical MRI, synthetic fMRI, and real life rs-fMRI data. The LICA approach is compared to several flavors of ICA. An important feature of LICA is that it is an unsupervised algorithm, where we do not need to postulate *a priori* information or models. This may allow its application as an exploratory data analysis procedure. LICA does not impose a probabilistic model on the sources. The lattice independence condition may be a more easily satisfiable restriction in order to find meaningful sources in data instances where ICA approaches can fail due to their statistical properties.

- In the application of LICA to the model-free (unsupervised) VBM analysis, we find a strong agreement between LICA results and those of ICA, and we can identify endmembers and sources that correspond closely to the significative detection of results in agreement with SPM and FSL, providing a validation of the approach. The problem with VBM and similar morphometric approaches is that we need to be able to give some interpretation to the findings of the ICA and LICA algorithms, that is, besides the obvious identification of voxels that correlate well with the indicative variable, the problem is to find additional regularities and give them some sense.

- The computational experiments performed on simulated fMRI data show that LICA performance is comparable to or improves over the ICA approach in the sense of discovering task-related sources and their spatial locations. Computational experiments on a well-known case study, have found that LICA gives results consistent with the gold standard provided by the SPM

approach. Compared to ICA algorithms, LICA reproduces better the SPM results.

- We compare LICA and ICA findings on rs-fMRI data in the form of detections based on the thresholding of the abundance images and mixing matrices. Both LICA and ICA are unsupervised approaches, so they do not force *a priori* assumptions on the localization of the findings, which must be interpreted after the analysis, risking to obtain results not in agreement with the expectations of the analysis. LICA detections are less sparse than those of ICA, but the medical assessment of findings is being carried out actually. The main quantitative conclusion of this study is that there is little agreement between LICA and ICA on this data. Moreover, when we consider the correlation of findings by LICA or ICA on the control versus the schizophrenic patient, we find that the LICA results show greater negative correlation than the results of ICA. We interpret this result as pointing to a greater capability to produce features for discrimination between control and patients based on resting state fMRI data. Anyway, we can not use ICA results as a validation reference, so validation of LICA results must rest on the medical expert assessment of its findings.

Future work Besides some other open theoretical questions, we want to state conveniently, and solve, the problem of finding the *right* number of LIS, and the *right* LIS. Those are non trivial problems in many other context (i.e. clustering), stated and solved as some kind of minimization problem. In our context, the problem is further complicated by the intrinsic non-linearity of endmember induction algorithms and the interleaving of the linear and non-linear procedures in LICA. It is not evident at this moment how to formulate a *well behaved* objective function for such purposes. Besides these fundamental problems, we will extend the validation evidence that may give the confidence to apply the method to new fMRI data sets as an exploratory tool by itself. We wishfully think that it could be applied to event oriented experiments, and to the task of discovering networks of activation in the brain.

Chapter 5

Lattice Computing Multivariate Mathematical Morphology

This chapter introduces the concept of Multivariate Mathematical Morphology based on the definition of a supervised ordering built on the Lattice Auto-associative Memories (LAAM) recall. Such constructions allow to set completely the Multivariate Mathematical Morphology in a Lattice Computing framework [**?**, **?**, **?**]. The chapter includes results on the application on resting state fMRI (rs-fMRI) data for the study of Schizophrenia patients with and without auditory hallucinations [**?**]. The content of the chapter is as follows: Section 5.1 gives an introduction to the chapter contents. Section 5.2 introduces the main ideas of Multivariate Mathematical Morphology. Section 5.3 provides the definition of the LAAM based reduced h-orderings. Section 5.4 provides the detailed experimental results over a specific rs-fMRI dataset. Section 5.5 gives some conclusions of this chapter.

5.1 Introduction

The extension of Mathematical Morphology to multivariate images needs to define appropriate orderings allowing to define the elementary morphological operators without introducing spurious results (i.e. false color). The general approach followed in this Thesis consists in the application of Lattice Auto-Associative Memories (LAAMs) [**?**, **?**] to the definition of a *LAAM-supervised ordering*, an specific kind of h-ordering [**?**], that allows the consistent definition of morphological operators on multivariate data. All the required calculations are defined using the Lattice

algebra operators ($\vee$, $\wedge$ and $+$) and therefore, LAAM-supervised ordering is faster and imposes less computational burden than the supervised orderings previously proposed in [**?**]. The LAAM-supervised *h*-function consists in the LAAM recall error for each voxel.

rs-fMRI images We consider rs-fMRI data of healthy controls (HC), schizophrenia patients with and without auditory hallucinations (SZAH and SZnAH, respectively) looking for brain network differences. Using the LAMM-supervised *h*-ordering is a Lattice Computing correspondent to the correlation [**?**] or independent component analysis [**?**] based approaches to the analysis of resting state fMRI searching for networks of low frequency synchronized components in the brain. In short, a seed voxel BOLD time series is used to build a LAAM, which is then applied to the remaining voxels of the brain fMRI 4D data. Here the *h*-function provides the functional similarity on which the identification of the brain networks is performed, and the map obtained from the whole brain volume is thresholded to detect functional connectivity. It can be also processed by morphological operators providing some specific features of the volume.

Experiments We report in this chapter three experiments performed on the resfMRI data. In the first two experiments, a group analysis is performed on the templates corresponding to each class of subjects computed by averaging their spatially normalized rsfMRI data. We inspect the Tanimoto coefficients between identified networks to decide the appropriate threshold value to report the detected brain functional networks.

1. In the first experiment, the reduced ordering function maps are used for the detection of brain networks. Specifically we compute the brain networks induced by specific brain sites, looking for differences in specific populations. Results show that the approach is able to find functionally connected cluster differences discriminating the subjects suffering auditory hallucination. Specifically, we perform experiments with background/foreground LAMM h-function.

2. In the second experiment, (a) we build templates for each population by averaging the registered 4D data, (b) we process the whole brain volume, (c) we focus on the background/foreground h-function map, (d) we follow the work in [**?**] exploring the network induced by an specific localization in the

brain, (e) threshold value is decided by inspection of the Tanimoto coefficients between the functional networks of each population class. Working on the population templates allows to assess group level effects that are visible on the average data.

3. In the third experiment, we report classification results on each population class. Data features for classification are obtained as follows: we build the LAAM-supervised h-function providing a reduced ordering map related to the left Heschl's gyrus. The Pearson correlation coefficient between the h-function values and the categorical variable at each voxel site allows to identify the most informative voxel sites. Feature vectors are constructed as the h-function values at these sites. Results with the baseline k-NN classifiers show that the approach can provide accurate discrimination between these populations.

5.2 Multivariate Mathematical Morphology

Mathematical Morphology was introduced by J. Serra [**?**, **?**] and P. Maragos [**?**] as a powerful tool for image analysis. Over the fifty years of its existence, Mathematical Morphology has proven to be useful in different fields, such as image processing, vision; and in different application: medical image segmentation, remote sensing classification, noise detection, texture analysis, shape recognition.

Morphological operations are mappings between complete lattices, denoted $\mathbb{L}$ or $\mathbb{M}$, that are partially ordered set where infimum and supremum are defined for all pairs of elements. The two elementary morphological operators are erosion and dilation. other operators and filters are defined as compositions of them. For every subset $Y \subseteq \mathbb{L}$ an *erosion* is a mapping $\varepsilon : \mathbb{L} \to \mathbb{M}$ that commutes with the infimum operation, $\varepsilon\left(\bigwedge Y\right) = \bigwedge_{y \in Y} \varepsilon\left(y\right)$. Similarly, a *dilation* is a mapping $\delta : \mathbb{L} \to \mathbb{M}$ that conmutes with the supremum operation, $\delta\left(\bigvee Y\right) = \bigvee_{y \in Y} \delta\left(y\right)$. O top of these basic operators it is possible to define image fiters such as the morphological gradient $g\left(Y\right) = \delta\left(Y\right) - \varepsilon\left(Y\right)$, or the top-hat $t\left(Y\right) = Y - \delta\left(\varepsilon\left(Y\right)\right)$.

5.2.1 Multivariate ordering

Morphological operators are well defined for scalar images, however their extension to multivariate images is not straightforward since defining a total order on

these vector spaces preserving some natural properties of the morphological operators is not easy. One salient property is that the result of erosion or dilation operators must be closed in the image range of values, i.e. no new colors are generated by their application. The proposals in [**?**, **?**, **?**, **?**] are examples of multivariate order definitions that suffer from the false color problem.

One way to accomplish that is the mapping of the multivariate values into a scalar through the definition of a reduced ordering [**?**, **?**]. A h-ordering is defined by a surjective mapping of the original data set onto a complete lattice $h : X \rightarrow \mathbb{L}$, so that the order in the target lattice induces a total order on the original data set X, that is, $r \leq_h r' \Leftrightarrow h(r) \leq h(r')$.

$$\mathbf{x} \leq_h \mathbf{y} \Leftrightarrow h(\mathbf{x}) \leq h(\mathbf{y}); \forall \mathbf{x}, \mathbf{y} \in X. \tag{5.1}$$

The reduced ordering can be defined on the basis of a supervised classifier trained with some pixel values extracted from the image. Discriminant function values or the estimated class *a posteriori* probabilities provide the surjective mapping h. Often, a two class discrimination between foreground and background classes is considered. In the formalization suggested in [**?**], a h-supervised ordering over a non-empty set X is a h-ordering satisfying the conditions $h(b) = \perp$, $\forall b \in B$, and $h(f) = \top$, $\forall f \in F$, where $B, F \subset X$ are subsets of X such that $B \cap F = \emptyset$, and $\perp$ and $\top$ are the bottom and top elements of the target lattice, respectively. Erosion operators increase image regions of points close to the background, and dilation operators will increase image regions of points close to the foreground.

As h-functions are not necessarily injective the induced h-ordering $\leq_h$ might be not a total order. When we need to differentiate among the members of the equivalence classes $\mathscr{L}[z] = \{\mathbf{c} \in \mathbb{R}^n | h(\mathbf{c}) = z\}$, the disambiguation criterion is the lexicographical order.

5.2.1.1 Multivariate morphological operators

The h-supervised erosion of a multivariate image $\{I(p) \in \mathbb{R}^n\}_{p \in D_I}$, where D_I is the spatial domain of the image, with structural object S, is defined as follows: $\varepsilon_{h,S}(I)(p) = I(q)$ s.t. $I(q) = \bigwedge_h \{I(s); s \in S_p\}$, where $\bigwedge_h$ is the infimum defined by the reduced ordering $\leq_h$ of Eq. (**??**), and S_p is the structural element translated to the pixel position p. The h-supervised dilation is defined as $\delta_{h,S}(I)(p) = I(q)$ s.t. $I(q) = \bigvee_h \{I(s); s \in S_p\}$, where $\bigvee_h$ is the supremum defined by the order of Eq. (5.1).

The morphological gradient for scalar valued images is computed as the difference between the dilations and erosions of image I with structuring element S: $g_S(I) = \delta_S(I) - \varepsilon_S(I)$. For multivariate images, the h-supervised morphological gradient can be defined as the h-supervised erosion $\varepsilon_{h,S}(I)$ and dilation $\delta_{h,S}(I)$ as follows:

$$g_{h,S}(I) = h(\delta_{h,S}(I)) - h(\varepsilon_{h,S}(I)). \tag{5.2}$$

5.3 LAAM-Supervised Ordering

5.3.1 LAAM's h-mapping

The LAAM h-mapping is defined as the Chebyshev distance between the original pattern vector and the recall obtained from the LAAM. In [**?**] this distance was used to define a lattice based Nearest Neighbor classifier. In our approach it performs the role of the supervised classifier. Formally, given a sample data vector $\mathbf{x} \in \mathbb{R}^n$ and a non-empty training set $X = \{\mathbf{x}_i\}_{i=1}^K$, $\mathbf{x}_i \in \mathbb{R}^n$ for all $i = 1, \ldots, K$, the LAAM h-mapping is given by:

$$h_X(\mathbf{c}) = d_C\left(\mathbf{x}^{\#}, \mathbf{x}\right), \tag{5.3}$$

where $\mathbf{x}^{\#} \in \mathbb{R}^n$ is the recalling response of dilative LAAM M_{XX} to the input of vector $\mathbf{x}$, i.e. $\mathbf{x}_M^{\#} = M_{XX} \boxslash \mathbf{x}$. The erosive memory W_{XX} recall, i.e. $\mathbf{x}_W^{\#} = W_{XX} \boxbslash \mathbf{x}$, could be used alternatively. Function $d_C(\mathbf{a}, \mathbf{b})$ denotes the Chebyshev distance between two vectors, given by the greatest absolute difference between the vectors' components: $d_C(\mathbf{a}, \mathbf{b}) = \bigvee_{i=1}^n |a_i - b_i|$.

5.3.2 Foreground LAAM h-supervised ordering

Given a training set X a Foreground LAAM h-supervised ordering, denoted by $\leq_X$, is defined on the LAAM h-mapping of Eq. (5.3) as follows:

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \mathbf{x} \leq_X \mathbf{y} \Longleftrightarrow h_X(\mathbf{x}) \leq h_X(\mathbf{y}). \tag{5.4}$$

The Foreground LAAM-supervised ordering generates a complete lattice $\mathbb{L}_X$, whose bottom element $\perp_X = 0$ corresponds to the set of fixed points of M_{XX} and W_{XX}, i.e. $h(\mathbf{x}) = \perp_X$ for $\mathbf{x} \in \mathscr{F}(X)$. On the other hand, the the top element is $\top_X = +\infty$.

5.3.3 Background/Foreground LAAM h-supervised orderings

In order to build a Background/Foreground LAAM h-supervised ordering, disjoint background B and foreground F training sets are required. The Foreground LAAM h-mapping defined in Eq. (5.3) is independently applied to the data using B and F as training sets, obtaining mappings h_B and h_F, respectively.We define a Background/Foreground (B/F) LAAM h-mapping $h_r(\mathbf{x})$ combining both h_B and h_F into an h-mapping as follows:

$$h_r(\mathbf{x}) = h_F(\mathbf{x}) - h_B(\mathbf{x}), \tag{5.5}$$

which is positive for $\mathbf{x} \in \mathscr{F}(B)$, and negative for $\mathbf{x} \in \mathscr{F}(F)$. Therefore, we assume it as a discriminant function such that $h_r(\mathbf{x}) > 0$ corresponds to pixels in the background class, and $h_r(\mathbf{x}) < 0$ to pixels in the foreground class. Points where $h_r(\mathbf{x}) = 0$ holds correspond to the decision boundary. The Relative h-supervised ordering, denoted $\leq_r$, is defined as follows:

$$\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, \mathbf{x} \leq_r \mathbf{y} \Longleftrightarrow h_r(\mathbf{x}) \leq h_r(\mathbf{y}). \tag{5.6}$$

The image of the B/F LAAM h-mapping is a complete lattice $\mathbb{L}_r$ whose bottom and top elements are $\bot = -\infty$ and $\top = +\infty$, respectively. Though this h-mapping does not fit strictly into the formalization proposed in [**?**] because $h_r(\mathbf{b}) \neq \bot$ for $\mathbf{b} \in B$ and $h_r(\mathbf{f}) \neq \top$ for $\mathbf{f} \in F$, the induced ordering and subsequent morphological operators provide good results.

5.4 Experimental results on rs-fMRI

The results reported in this section are explorations over resting state fMRI data obtained from a 28 healthy control subjects (NC), and two groups of schizophrenia patients: 26 subjects with and 14 subjects without auditory hallucinations (SZAH and SZnAH respectively), selected from an on-going study in the McLean Hospital, Boston, Ma. Details of image acquisition and demographic information are given in the Appendix of this Thesis. Before going into the detail of the experiments, we provide relevant definitions.

Tanimoto coefficient We use the Tanimoto coefficients computing the similarity between detected networks to decide the appropriate threshold to apply in order to

report the network detection. Given a pair of sets A and B, the Tanimoto coefficient measures their similarity as the ratio between the cardinalities of their intersection and union:

$$T(A,B) = \frac{|A \cap B|}{|A \cup B|}, \tag{5.7}$$

that is, the it measures the relative extent of their overlapping. The Tonimoto coefficient is normalized $T \in [0,1]$. Complete dissimilarity corresponds to $T = 0$, identity corresponds to $T = 1$. When comparing image segmentation results, the sets are image regions.

Pearson Correlation The Pearson correlation coefficient is given by:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{\left[n\sum x^2 - (\sum x)^2\right]\left[n\sum y^2 - (\sum y)^2\right]}} \tag{5.8}$$

where $r \in [-1,1]$, $r = 1$ means that two variables have perfect positive correlation and $r = -1$ means that there is a perfect negative correlation between them. In our case Pearson correlation evaluates the nexus between *a priori* known class labels and fMRI neural connectivity by means of h-function.

5.4.1 Experiment 1

The aim of the experiments in this section looks for brain networks that may allow the discrimination of healthy control subjects, schizophrenia patients with and without auditory hallucinations. The results show that the LAAM-supervised h-orderings detect quite different brain networks depending on the subject using the same h-function built from selected voxel seeds. Network localizations correlated with an specific voxel, preferentially from the auditory cortex will show some effect related to the auditory hallucinations. The voxel time series used to build the LAAM is extracted from one healthy control subject data. We compute the map corresponding to the application of the LAAM-supervised h-function to each voxel in one healthy control subject and one schizophrenia patients, showing the network constituted by most similar voxels according to the h-function. Finally, we visualize the detection obtained by the peaks of the top-hat transformation. We present results over the brain mappings given by the one-side and background/foreground orderings.

One-side supervised h-ordering We have tested the results of two seeds, one in the frontal lobe and other in the auditory cortex. Figure 5.1(a) shows the location of the voxel seed near the frontal lobe. It has been picked arbitrarily in the gray matter. Figure 5.1(b) shows the network located with this seed voxel in the control subject data, without any further postprocessing. Figure 5.1(c) shows the network in the patient with auditory hallucinations. A number of related voxels appear larger than in the control subject and the patient without hallucination. Figure 5.1(d) shows the same detection result on the patient without hallucinations. It can be appreciated that the network is much smaller than in the AH case.

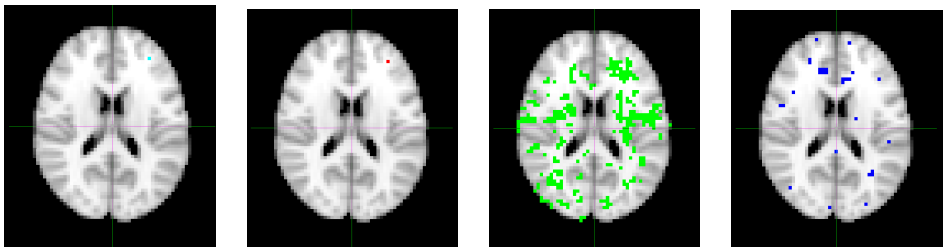
Figure 5.2(a) shows the location of the voxel seed in the auditory cortex. Remaining subfigures have the same meaning as in figure 5.1. Observe that the health control does not have any induced network in this case. Again, the network of the patient with hallucinations is very large.

Therefore proposed approach provides brain networks which are dependent on the voxel seed. The main feature of the results is that, independently of the seed voxel used to build h_X, we find a strong discordance between the network of Schizophrenic patients with and without AH. This conclusion is confirmed by the top-hat filter localizations shown in figure 5.4(a) and 5.4(b) computed on the h-map induced by the seed shown in figure 5.1(a) 5.2(a).

Background/foreground supervised ordering We have selected as a background seed the voxels of WM in the Temporal Lobe and CSF in the ventricles, and as foreground voxel of GM: one in the Auditory Cortex and another in Frontal Lobe, to compute the h_r map. The seed selection is shown in figure 5.3(a)-(d). The meaning of the remaining images in figure 5.4 is the same as in previous images. Again, we find a strong difference between the network locations detected in the three subjects from the same map. This conclusion is reinforced by the top-hat filter localizations shown in figure 5.5(a)-(d) computed on the h-map induced by the seed shown in figure 5.3(a)-(d).

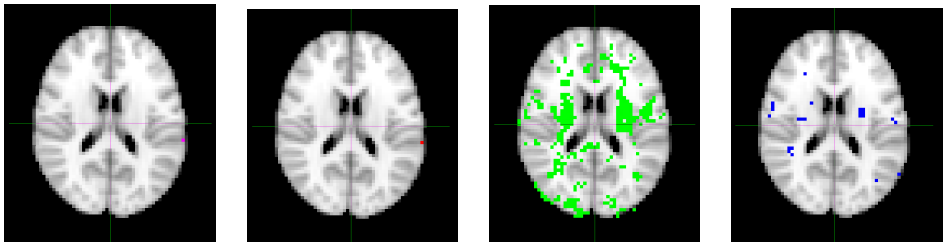
5.4.2 Experiment 2

After the preprocesing step detailed in the Appendix A, we compute the Z-scored time series at each voxel. These normalized images were used to calculate the time-average of BOLD across subjects for each group. Finally, we compute the 4D group-average of subjects from HC, SZAH and SZnAH groups to build corre-



(a) (b) (c) (d)

Figure 5.1: Seed from the frontal lobe. (a) location of the seed voxel in the healthy control volume, (b) network of corresponding voxels in the healthy control, (c) schizophrenia patient with auditory hallucinations, and (d) schizophrenia patient without auditory hallucinations.



(a) (b) (c) (d)

Figure 5.2: Seed from the auditory cortex. (a) location of the seed voxel in the healthy control volume, (b) network of corresponding voxels in the healthy control, (c) schizophrenia patient with auditory hallucinations, and (d) schizophrenia patient without auditory hallucinations.

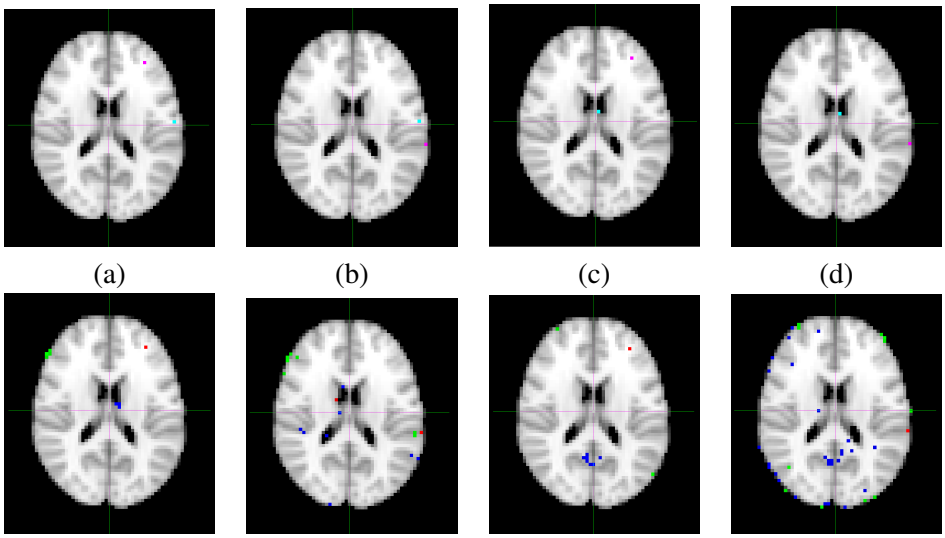
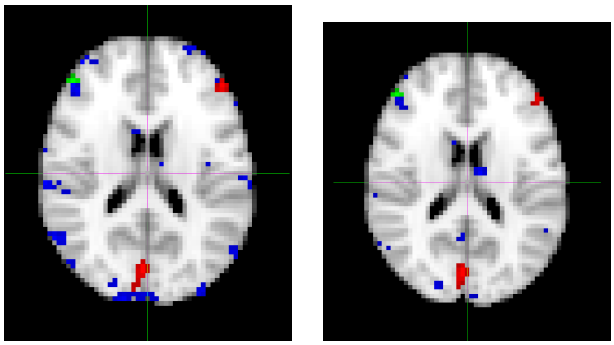
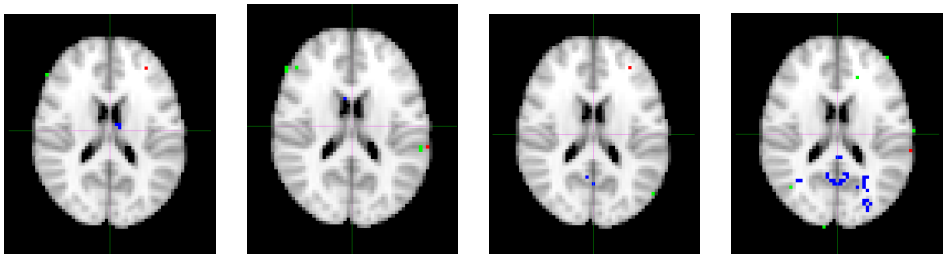


Figure 5.3: Two voxel seeds: (a) background from WM and foreground from GM of Frontal Lobe, (b) background from WM anfd foreground from GM of Auditory Cortex, (c) background from CSF of the ventricule and foreground from GM of Frontal Lob, (d) background from CSF of the ventricule and foreground from GM of Auditory Cortex, – used to build the h_r ordering. Blue and pink colors indicate the back- and fore-ground voxels. (e) location of the seed voxels in the healthy control volume, (f) network of corresponding voxels in the healthy control, (g) schizophrenia patient with auditory hallucinations, and (h) schizophrenia patient without auditory hallucinations.



(a) (b)

Figure 5.4: Top-hat localizations computed on (a) and (b) the h_X ordering induced by the seed in figure 5.1(a) and 5.2(a). Red, green, blue voxel colors correspond to healthy control, schizophrenia no auditory hallucination, and schizophrenia no auditory hallucination, respectively..



(a) (b) (c) (d)

Figure 5.5: Top-hat localizations: the h_r ordering induced by the pair of backgrounf/foreground seeds in figure 5.3(a)-(d).

sponding templates for each population.

Identification of relevant networks According to the findings reported in [**?**], our experiments aim to obtain network localizations correlated with an specific voxel placed on the left Heschl's gyrus (LHG; MNI coordinates -42,-26,10) whose localization is illustrated in figure 5.6, preferentially from the auditory cortex in order to ascertain some effect related to the auditory hallucinations. The voxel time series used as the seed to build the LAAM corresponding to the foreground is extracted from this MNI coordinate from the template data of each group. We compute the map corresponding to the application of the Foreground/Background LAAM h-function on each template using as foreground and background training data the seed voxels identified in figures 5.6 and 5.7, respectively. The foreground voxel seed corresponds to the LHG, and the background voxel to the CSF in one of the ventricles, which corresponds to irrelevant or noisy time series.

Computing the Foreground/Background LAAM h-function produces a real valued map over the brain volume, where functional networks are identified applying a threshold to this map. We would like to set this threshold so that the differences between the networks from each population are greatest while the size of the detected network sites (aka voxel clusters) are also greatest. To this end we compute the Tanimoto coefficient between the brain regions identified in different templates when setting a threshold value on their respective h-mappings. In fact, we deal with sets of voxels sites, which depend on the value of the threshold applied to the h-function map. Therefore, $X(\theta)$ is the set of voxel sites with h-function above the θ threshold. Consequently, we denote $T(\theta) = T(X(\theta), Y(\theta))$. We are looking for the existence of discriminant regions, that is, brain networks that are unique to each population. Therefore, we look for small values of the Tanimoto coefficient, while the size of the identified regions (brain networks) remains significant.

Results Figure 5.8 shows two plots. The first plot corresponds to the evolution of pairwise population network similarity measured by Tanimoto coefficients $T(\theta)$, increasing the threshold value. It can be appreciated that the similarities of the HC versus any of the two patient populations are very much the same. However, the Tanimoto coefficient of the two patient populations are significantly higher, confirming the intuition that they share many functional network traits. However, we are interested here in finding differences that may be useful for discrimination/classification. Therefore, we focus in the lower values: for $\theta > 0.7$, Tanimoto

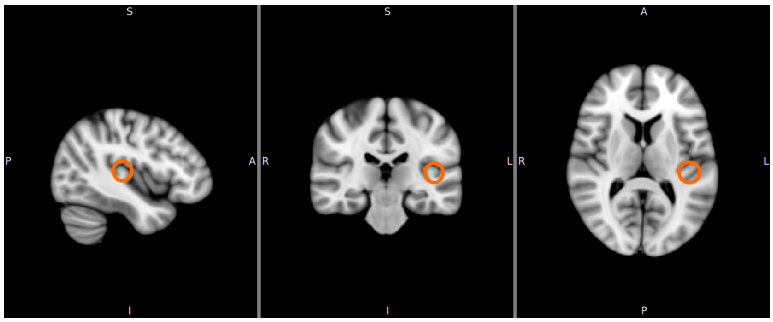


Figure 5.6: Foreground voxel seed site from the left Heschl's gyrus (LHG; -42,-26,10).

coefficients $T(\theta)$ comparing healthy and patients are close to zero , meaning that the networks are almost disjoint. The comparison between patient classes gives almost zero Tanimoto coefficients for $\theta > 0.8$. The second plot corresponds to the size of the networks, measured as the cardinality of the sets of voxel sites. It can be appreciated that for $\theta > 0.7$ the networks become very small, therefore, we have chosen $\theta = 0.7$ for the figures below.

Figure 5.9 shows the networks found related to the LHG seed voxel for the (a) healthy control (HC), (b) schizophrenic with auditory hallucinations (SZAH), and (c) schizophrenic without auditory hallucinations (SZnAH). Notice that the size of the SZAH network is bigger and they are more spread than the SZnAH network. There is a clear difference relative to the HC network. We think that these sites can be proposed as specific biomarkers. We have found that some of them are in agreement with previous reported findings [**?**], though the exhaustive listing of detection will be given elsewhere. We show in Figure 5.10(a) the intersection between the SZnAH and SZAH networks, which is very small but significant, and in Figure 5.10(b) the difference network corresponding to the clusters active only in SZnAH. There are many voxel sites which may be taken as biomarkers to discriminate between patient these classes. For a global view of the extent of the discriminant regions, we show a 3D visualization in Figure 5.11.

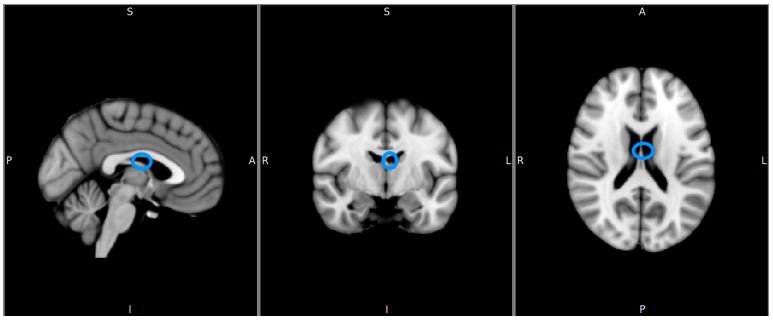
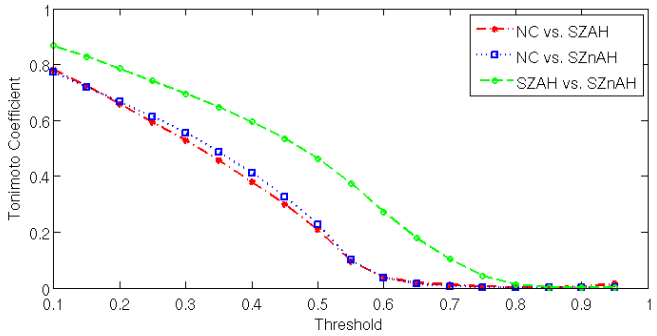
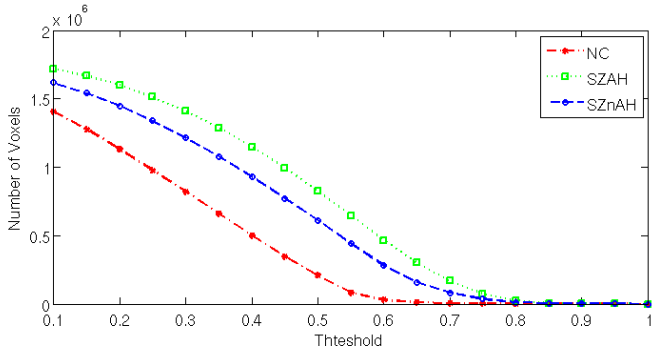


Figure 5.7: Background voxel seed site from CSF of the ventricle.



(a)



(b)

Figure 5.8: Effect of threshold value on the identified networks on background/foreground h-function brain map. (a) Tanimoto Coefficient comparing networks from each pair of population, and (b) size of the detected clusters.

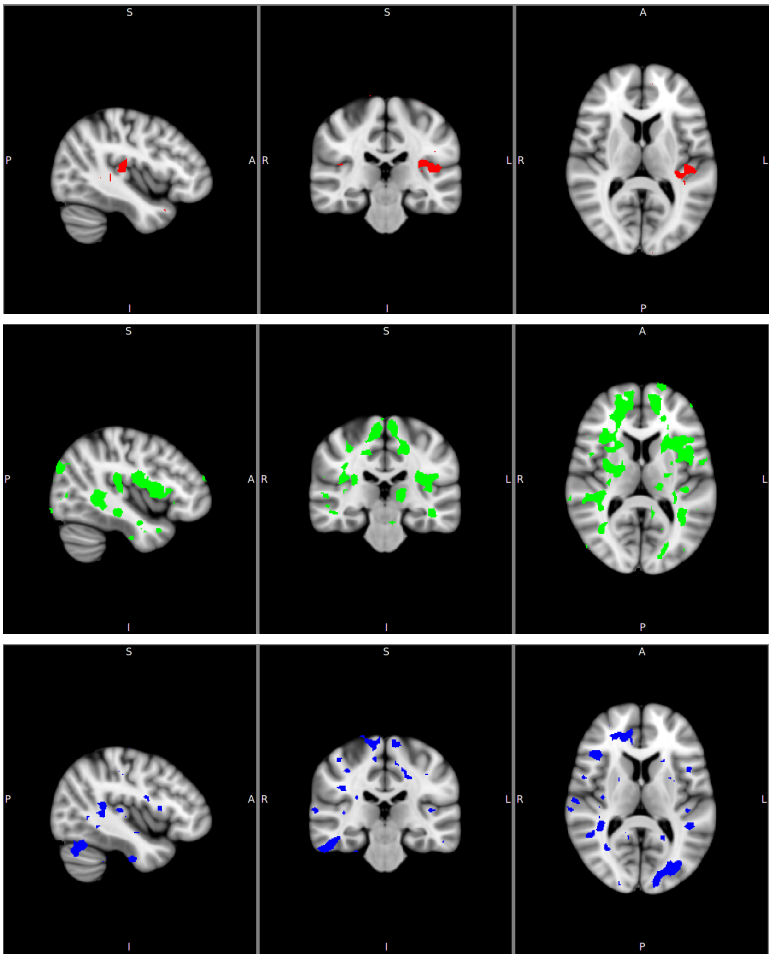
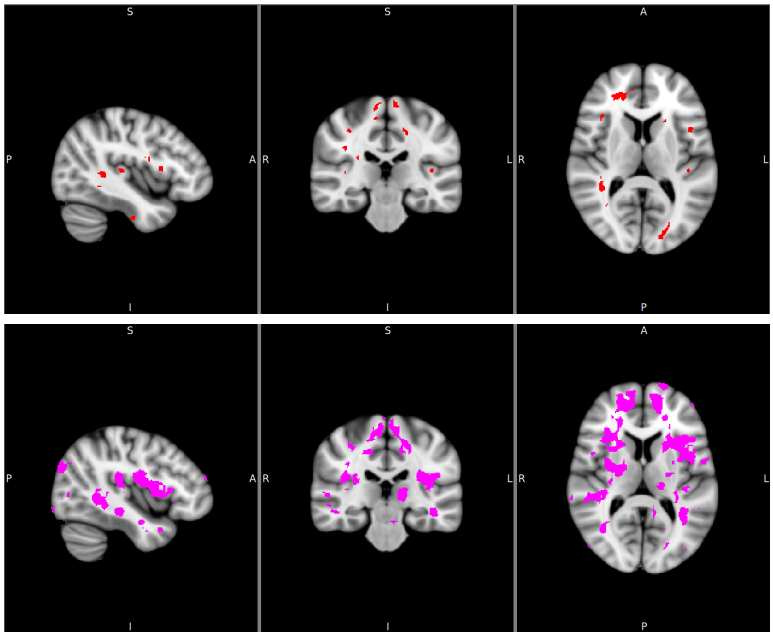


Figure 5.9: Networks identified by thresholding the Background/Foreground h-function induced by the pair of background/foreground seeds in figure 5.6 and 5.7 (a) healthy controls (HC), (b) schizophrenics with hallucinations, (c) schizophrenics without hallucinations.



(a)

(b)

Figure 5.10: Comparison of networks obtaining by thresholding background/foreground h-functions on the templates of the two types of schizophrenia patients (with and without auditory hallucinations): (a) the intersection network, (b) the network appearing only on the template of patients with hallucination (SZAH)

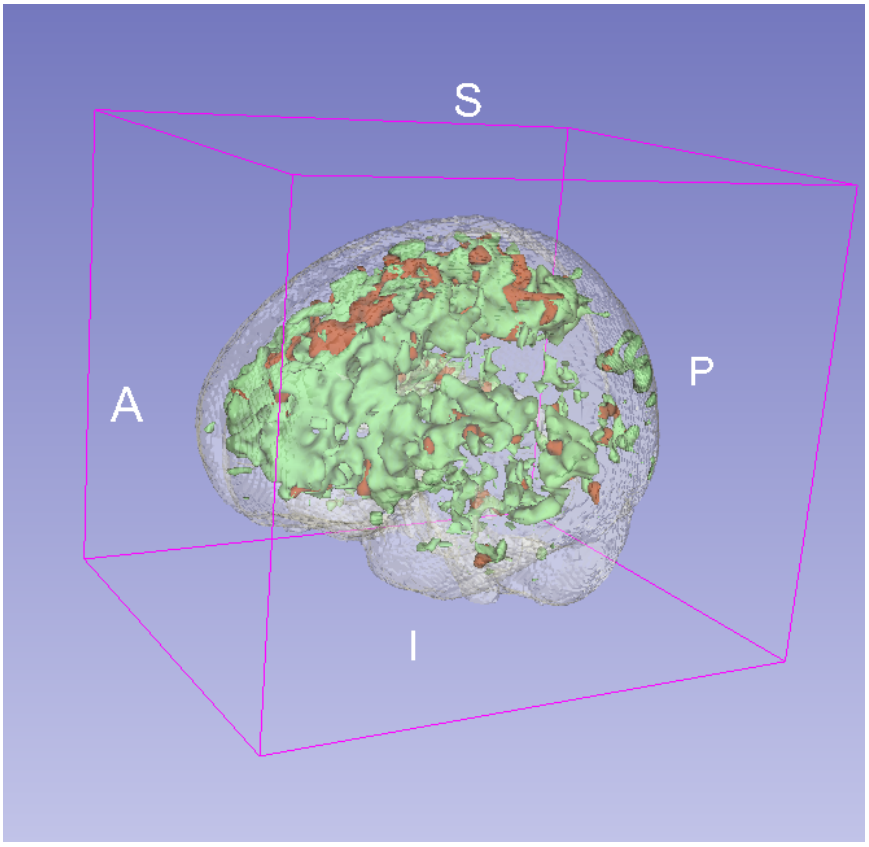


Figure 5.11: 3D visualization of the brain networks appearing only in the SZAH population template (green), and the common networks between SZAH and SZnAH populations (brown).

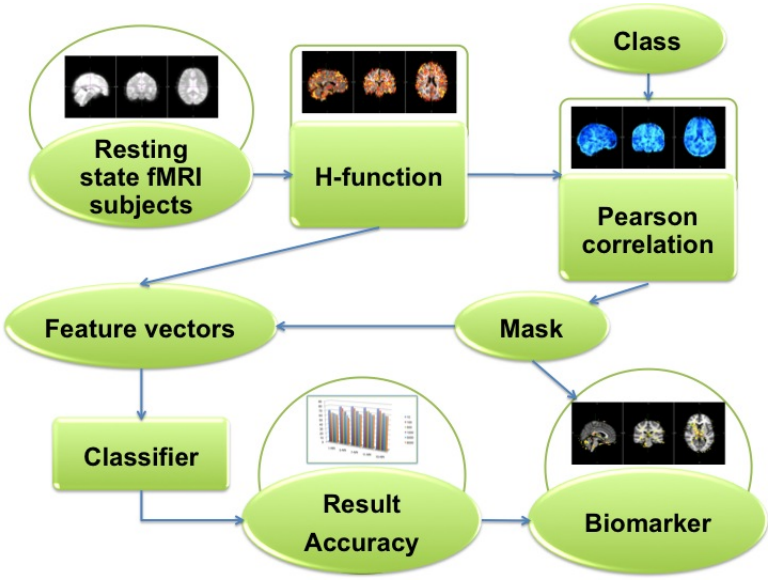


Figure 5.12: Pipeline of our experimental design

5.4.3 Experiment 3

Experimental pipeline Figure 5.12 shows the graphical description of our experimental process. Resting state fMRI data is first preprocessed to ensure that all fMRI volumes are aligned and warped to the spatially normalized structural T1-weighted data. On the normalized data, we compute the B/F LAMM h-mapping where the Background data corresponds to CSF in the brain ventricle voxels, and the Foreground data is a selection of voxels in the LHG according to [**?**]. The *h*-mappings are used to compute across volume voxel-wise Pearson correlation with the categorical variable specifying the class of the subject, obtaining a correlation map. Selection of the voxels sites with greatest absolute value of correlation coefficients defines the masks for feature extraction, which are used to build the feature vectors from the individual *h*-maps. These masks are providing localizations for image biomarkers that may have biomedical significance, therefore we report them separately. Feature vectors are used to perform classification experiments, applying a 10-fold cross validation methodology. We use k-NN classifiers to provide baseline results. Accuracy results are assumed to provide some endorsement of the value of the image biomarkers identified by the feature masks.

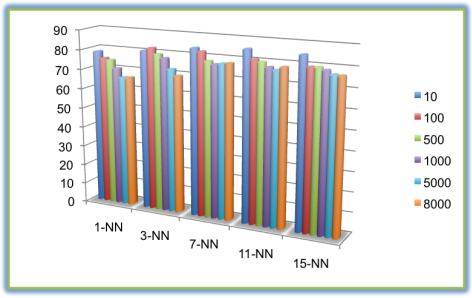
Classification results Figure (5.13) shows the results of the classification experiments on the discrimination the possible pairs of classes: Healthy controls versus Schizophrenia patients (HC vs. Schiz), versus patients without auditory hallucinations (HC vs. nAH), with auditory hallucinations (HC vs. AH), and between classes of patients (nAH vs. AH). The color bars identify the size of the feature vectors, which are built from voxels sites with greatest absolute Pearson's correlation coefficients. In all cases, classification performance decreases with the largest sizes of the feature vectors, which is to be expected because the k-NN classifier suffers from the curse of dimensionality. The best results are obtained in the (HC vs. nAH) case, suggesting that these kind of patients could be better discriminated from healthy controls. Discrimination of the auditory hallucination (nAH vs. AH) is not successful, however we expect that further experimentation will improve results.

Feature localization in the brain Figure (5.14) shows the voxel sites of the feature extraction in the above enumerated cases. These localizations may server as biomarkers for additional research.

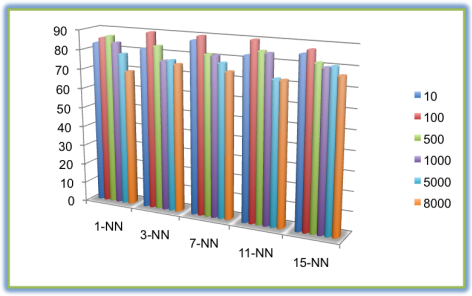
5.5 Conclusions

We introduce in this chapter a Multivariate Mathematical Morphology using lattice computing techniques. Specifically, using the LAAM reconstruction error measured by the Chebyshev distance as a reduced ordering h-map, we define several h-supervised orderings and ensuing mathematical morphology operators and filters. Specifically, we introduce a Foreground and a Foreground/Background/ LAAM-supervised orderings. The main benefit of these definitions is the consistency of the morphological operators and filters. Proposed method for fMRI data analysis needs the definition of seed voxels. However this method does not involve any common statistical techniques and assumptions, being model-free in a very extensive point of view. Moreover the method relies only in lattice computing operators, so that the only operations required for its intelligent wandering are min, max and addition which introduce less error than other arithmetic approaches.

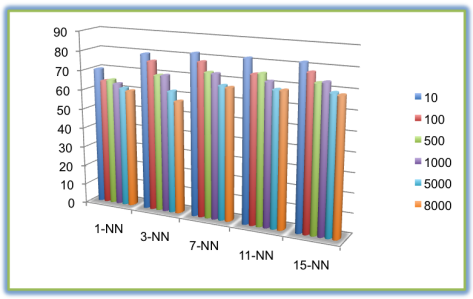
Regarding rs-fMRI, we give results of three experimental settings on the identification of differences between brain networks of schizophrenic patients with and without auditory hallucinations, which may allow to define discriminating voxel sites for feature extraction. In the first experiment, the basic approach shows that



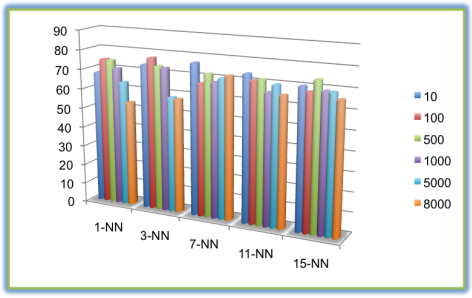
HC vs. Schiz



HC vs. nAH

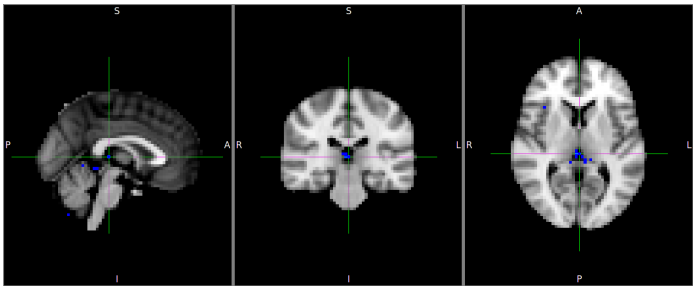


HC vs. AH

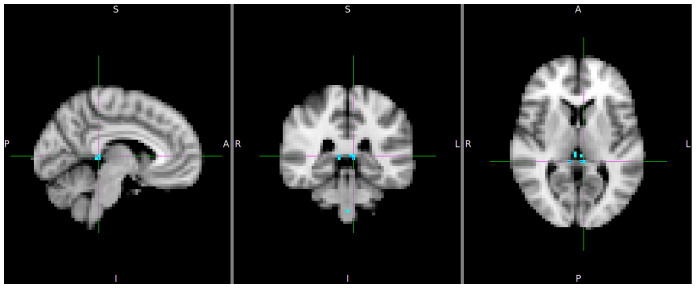


nAH vs. AH

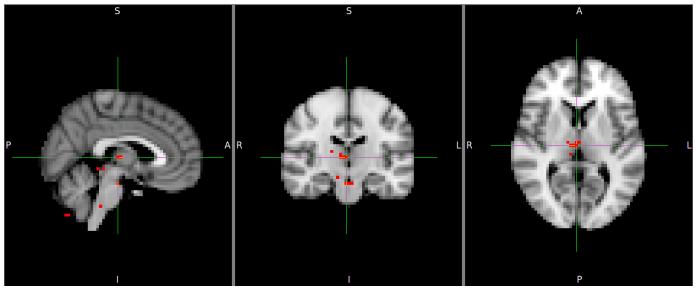
Figure 5.13: Maximum Classifier Accuracy found in 10 repetition of 10-fold cross validation for k-NN classifier $k = 1, 3, 7, 11, 15$. The bar colors represent diferent number of extracted features.



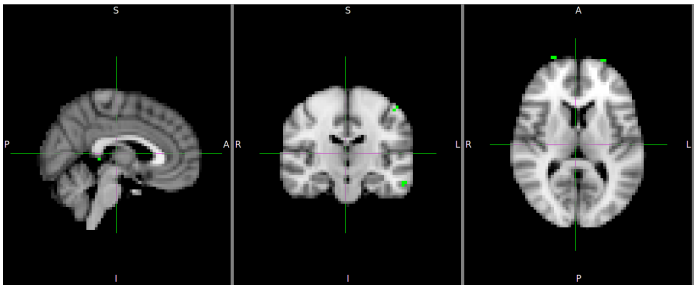
HC vs. Schiz



HC vs. AH



HC vs. nAH



nAH vs. nAH

Figure 5.14: Visualization of Localization

the approach is able to produce some network localizations with strongly different according to the subject class, inducing some kid of discrimination that are afterwards exploited by machine learning approaches.

In the second experiment, we compute brain networks setting a threshold on the h-mappings computed using appropriate foreground (LIH gyrus) and background (CSF) voxel seeds for the training of the LAMM. The population study is performed on the average templates per population, i.e. we build average templates for healthy patients, schizophrenia patients with and without auditory hallucination. The identified networks have clear differences between populations, which may allow to discriminate between them on an individual basis, using some classification approach.

In the third experiment, the Foreground/Background/ LAAM-supervised h-map on resting state fMRI provides the features for the identification of potential biomarkers for schizophrenia and variants with and without auditory hallucinations by computing the Pearson's correlation coefficient with the categorical variable. These biomarkers are evaluated in the terms of the corresponding classification accuracy achieved on the feature vectors extracted from the selected voxel sites. We find that the classification results are encouraging, with best results obtained in the discrimination between healthy controls and patients without auditory hallucinations. Further results will be obtained applying more sophisticated classifier systems to the data. Application of morphological filters to perform feature selection is also considered on the basis of the well defined Multivariate Mathematical Morphology.

Appendix A

Data

This Appendix is devoted to the description of some of the datasets that have been used in the experimental works along the thesis. From the medical point of view we have treated imaging data from several diseases: Alzheimer's Disease, Schizophrenia, and Abdominal Aortic Aneurysm. The latter datasets have been accessed through the collaboration with Josu Maiora, which obtained them for his PhD work. The contents of the Appendix are as follows: Section A.1 provides some medical background on each disease. Section A.2 introduces some of the image modalities used in the computational experiments. Section A.3 gives a short review of Voxel Based Morphometry. Section A.4 contains the description of the anatomical feature dataset for Alzheimer's Disease extracted from the OASIS database. Section A.5 presents the resting state data used for the study on Schizophrenia patients. Section A.6 provides the description of the Abdominal Aortic Aneurysm database.

A.1 Medical background

A.1.1 Alzheimer's Disease

Alzheimer's Disease (AD) is a neurodegenerative disorder, which is one of the most common cause of dementia in old people. Currently, due to the socioeconomic importance of the disease in occidental countries it is one of the most studied. The diagnosis of AD can be done after the exclusion of other forms of dementia but a definitive diagnosis can only be made after a post-mortem study of the brain tissue. This is one of the reasons why Magnetic Resonance Imaging

(MRI) based early diagnosis is a current research hot topic in the neurosciences. The pharmaceutical companies have already recognized that imaging techniques especially MRI and Positron Emission Tomography (PET) provide "surrogate" information concerning the pattern and rate of neurodegeneration, which can be used to monitor the effects of treatments which slow the progression of neurodegeneration. Therefore, there is high interest in the development of automated detection procedures based on MRI and other medical imaging techniques.

A.1.2 Schizophrenia

Schizophrenia is a brain disorder of mental activity that affects abilities of the person in processing the information, giving rise to difficulty understanding of the surrounding world. Schizophrenia is characterized by a suit of symptoms. These include auditory hallucinations, paranoid delusions, lack of emotion, disorganized and slow thinking, difficulty understanding. Schizophrenia is a severe psychiatric disease that is characterized by delusions and hallucinations, loss of emotion and disrupted thinking. In the last twenty years, there has been an explosion of knowledge about MRI analysis and how healthy brains and in brains with schizophrenia work, but little is known for sure. Schizophrenia is still a hermetic and difficult to understand disease, but reserch is not static and trying to develop new methods and tools of comprehension the schizophrenia.

A.1.3 Aortic Abdominal Aneurysm

An aneurysm is a focal dilation of a blood vessel to more than twice its normal diameter. Aneurysms are most commonly found in large arteries (aorta, iliac, and femoral); however, they have been reported in smaller arteries such as the radial or coronary arteries as well. The etiology of aneurysm is currently believed to be multi-factorial with atherosclerosis contributing the greatest part to the disease process. Other causes may include infectious etiologies, traumatic injury, chronic lung diseases, genetic disorders, smoking, and bio-mechanical factors such as hypertension, disturbed blood flow, and wall tissue degradation. The prevalence of aneurysms is greatest in the infra-renal abdominal aorta.

Abdominal Aortic Aneurysm (AAA) [**?**] is a dilation of the aorta that occurs between the renal and iliac arteries due to weakening of the vessel wall. The weakening of the aortic wall leads to its deformation and the generation of a thrombus. If the aneurysm gets too big, it can break. If left untreated, nearly all aneurysms

continue to enlarge and eventually rupture. Aneurysm with a diameter of 5 cm or greater should be treated. The rupture of an aneurysm can have very serious consequences and even cause death.

A.2 Medical Image Modalities

A.2.1 fMRI

Noninvasive techniques can measure cerebral physiologic responses during neural activation. Among them, fMRI [**?**] uses the blood oxygenation level dependent (BOLD) contrast to detect physiological alterations, such as neuronal activation resulting in changes of blood flow and blood oxygenation. The signal changes are related to changes in the concentration of deoxyhemoglobin, which acts as an intravascular contrast agent for fMRI. Most of the fMRI examinations are performed using T2* weighted gradient echo pulse sequences. The various fMRI-methods have a good spatial and temporal resolution, limited only by the delays with which the autoregulatory mechanisms of the brain adjust blood flow to the metabolic demands of neuronal activity. Since these methods are completely noninvasive, using no contrast agent or ionizing radiation, repeated single-subject studies are becoming feasible [**?**].

An fMRI experiment consists of a functional template or protocol (e.g., alternating activity and rest periods for a certain time) that induces a functional response in the brain. The aim of the experiment is to detect the response to this time varying stimulus, through the examination of the signal resulting from the BOLD effect, in defined volume elements (voxels). The functional information of a voxel has to be extracted from its BOLD signal time series. One fMRI volume is recorded at each sampling time instant during the experiment. The time sampling frequency is determined by the resolution of the fMRI pulse sequence. The complete four-dimensional dataset (three dimensions in space, one dimension in time) consists of subsequently recorded three-dimensional (3-D) volumes. The acquisition of these functional volumes runs over periods lasting up to several minutes. These long acquisition times allow for the introduction of motion artifacts that must be corrected through careful volume registration procedures.

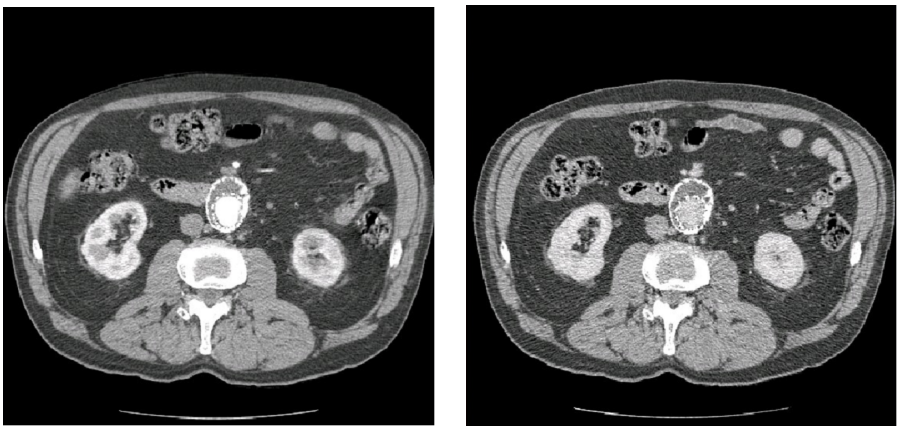
The most extended analysis approach for fMRI signals is the Statistical Parametric Maps (SPM) [**?**, **?**], which has evolved into a free software package. This method consists in the separate voxel estimation of the regression parameters of

General Linear Model (GLM). In GLM, data is explained as a linear combination of regressors defined of the basis of *a priori* knowledge about the data and the problem, which conform the so-called design matrix. The linear combination coefficients are estimated by conventional least squares methods. The design matrix is built corresponding to the experimental design. A contrast is then defined on the estimated regression parameters, which can take the form of a t-test or an F-test. The theory of Random Fields is then applied to correct the test thresholds, taking into account the spatial correlation of the independent test results.

Application of Independent Component Analysis (ICA) to fMRI [**?**, **?**, **?**] assumes that the time series observations are linear mixtures of independent sources which can not be observed. Reports on the research application of ICA to fMRI signals include the identification of signal types (task related and physiology related) and the analysis of multisubject fMRI data. The most common approach is the spatial ICA that looks for spatial disjoint regions corresponding to the identified signal types. It has been claimed that ICA has identified several physiological noise sources as well as other noise sources (motion, thermodynamics) identifying task related signals. Diverse ICA algorithms have been tested in the literature with inconclusive results. Among the clinical applications, ICA has been used to study the brain activation due to pain in healthy individuals versus those with chronic pain [**?**], the discrimination of Alzheimer's patients from healthy controls [**?**], the classification of schizophrenia [**?**] and studies about the patterns of brain activation under alcohol intoxication [**?**].

A.2.2 Resting state fMRI

Resting state fMRI (rsfMRI) data has been used to study the functional connectivity in the brain [**?**, **?**, **?**], looking for temporal correlation of low frequency oscillations in diverse areas of the brain, which provides a kind of brain functional fingerprint. Because the subject is not performing any explicit cognitive task, the functional network is assumed as some kind of brain fingerprint, which can be used to detect biomarkers of cognitive or neurodegenerative diseases. Resting state fMRI experiments do not impose constraints on the cognitive abilities of the subjects. For instance in pediatric applications, such as the study of brain maturation [**?**], there is no single cognitive task which is appropriate across the aging population. Resting state fMRI has being found useful for performing studies on brain evolution based on the variations in activity of the default mode network [**?**], depression (using



(a) (b)

Figure A.1: Comparison of vessel intensity values between CT and CTA slice. a) in CTA slice using the contrast agents, blood in lumen is highlighted for a better view. b) in CT slice without using the contrast agent, intensity values of lumen and thrombus are similar.

regional homogeneity measures) [**?**], Alzheimer's Disease [**?**], and schizophrenia. Computational approaches applied include hierarchical clustering [**?**], independent component analysis (ICA) [**?**, **?**, **?**], fractional amplitude of low frequency analysis [**?**], multivariate pattern analysis (MVPA) [**?**, **?**].

A.2.3 Computed Tomography Angiography

Computed Tomography (CT), a modality for non-invasive medical imaging has been established as the gold standard in many areas. Computed Tomography (CT) is a further development of the traditional X-ray technique. It is a medical imaging method employing tomography to generate a large series of cross-sectional images of the body. Though the CT slice is a two-dimensional (2D) image, using image processing, anatomical structures can be segmented and three-dimensional (3D) rendering can be created, which enhances the visual information of planar images and give physicians anatomical information of the region of interest .

Early CT scanners could only acquire images as a single axial slice at a time (sequential scanning), Later, several generations of scanners have been developed. Spiral CT scanners enables the X-ray tube to rotate continuously in one direction around the patient. Multi-slice CT machines (now up to 64-slice) utilizes the prin-

ciple of the spiral scanner and incorporate multiple rows of detector rings. The latest generation of CT scanner is Dual Source CT scanners, which were introduced in 2005. They allow higher temporal and spacial resolution, therefore reducing motion blurring at high heart rates and potentially requiring a shorter breath-hold time.

Volumetric data acquired by the modern CT scanner can be reconstructed to suit most clinical requirements. The data can be reformatted in various planes which is called multiplanar reformatting (MPR). We can get sagittal, coronal and oblique plane views along with the standard trans-axial plane. The stack of 2D slices can also be reconstructed as volumetric (3D) representations of structures for a better anatomical view. Contrast between different tissues of the body can be improved by the use of different contrast agents. These contrast agents are used to highlight specific areas so that the organs, blood vessels, or tissues are more visible. A.1 highlights the differences in contrast due to the use of a blood contrast. Contrast-enhanced Computed Tomography Angiography (CTA) is the most widely used medical imaging technology for getting exact knowledge of the position, shape, size of an aneurysm and the occurrence of endoleaks.

A.2.4 CTA for AAA

CTA allows minimally invasive visualization of the Aorta's lumen, thrombus and calcifications. The segmentation of the AAA thrombus is a challenging task due to the low thrombus signal contrast, great shape variability, both intra and inter-subjects, and little availability of prior information. Specifically, this is a blood vessel segmentation problem [**?**, **?**, **?**].

Classification based approaches to AAA thrombus segmentation allow to learn the optimal segmentation from the data. Some examples in the literature: A classification approach that needs an initial manual segmentation of the Aorta lumen [**?**]; an active shape model that uses the classification of grayscale profiles to move the active contour [**?**]; Support Vector Machine (SVM) used to drive a level set segmentation [**?**]; a Gaussian Mixture Model probability map drives a deformable NURBS model in [**?**]; finally, a segmentation based on voxel classification using Random Forests is proposed in [**?**, **?**].

AAA thrombus segmentation methods reported in the literature need a lot of human interaction or *a priori* information one way or the other. In [**?**] an initial rough specification of the aneurysm surface is refined by means of level set seg-

mentation driven by an *a priori* model and the likelihood estimation provided by Support Vector Machine classifiers trained on voxel location, intensity and texture features. In [**?**] a deformable NURBS model is driven by a probability map built from a Gaussian Mixture Model trained on selected samples. This approach needs an initial manual lumen segmentation and intensity renormalization to avoid convergence mishaps of the NURBS model adaptation. In [**?**] the AAA thrombus after endovascular repair is detected following a radial model approach needing the specification of the lumen centerline and some manually tuned correction performed on the polar coordinate representation of the image. A graph-cut approach constrained by a geometrical model is proposed in [**?**], needing a previous lumen segmentation and centerline computation. The approach iterates labeling and geometric model re-estimation, which are costly processes.

A.3 Voxel-based Morphometry (VBM)

Morphometry analysis has become a common tool for computational brain anatomy studies. It allows a comprehensive measurement of structural differences within a group or across groups, not just in specific structures, but throughout the entire brain. Voxel Based Morphometry (VBM) is a computational approach to neuroanatomy that measures differences in local concentrations of brain tissue, through a voxel-wise comparison of multiple brain images [**?**]. For instance, VBM has been applied to study volumetric atrophy of the grey matter (GM) in areas of neocortex of AD patients vs. control subjects [**?**, **?**, **?**]. The processing pipeline of VBM is illustrated in figure A.2. The procedure involves the spatial normalization of subject images into a standard space, segmentation of tissue classes using *a priori* probability maps, smoothing to correct noise and small variations, and voxel-wise statistical tests. Smoothing is done by convolution with a Gaussian kernel whose the Full-Width at Half-Maximum (FWHM) is tuned for the problem at hand. Statistical analysis is based on the General Linear Model (GLM) to describe the data in terms of experimental and confounding effects, and residual variability. Classical statistical inference is used to test hypotheses that are expressed in terms of GLM estimated regression parameters. This computation of given contrast provides a Statistical Parametric Map (SPM), which is thresholded according to the Random Field theory to obtain clusters of significant voxels.

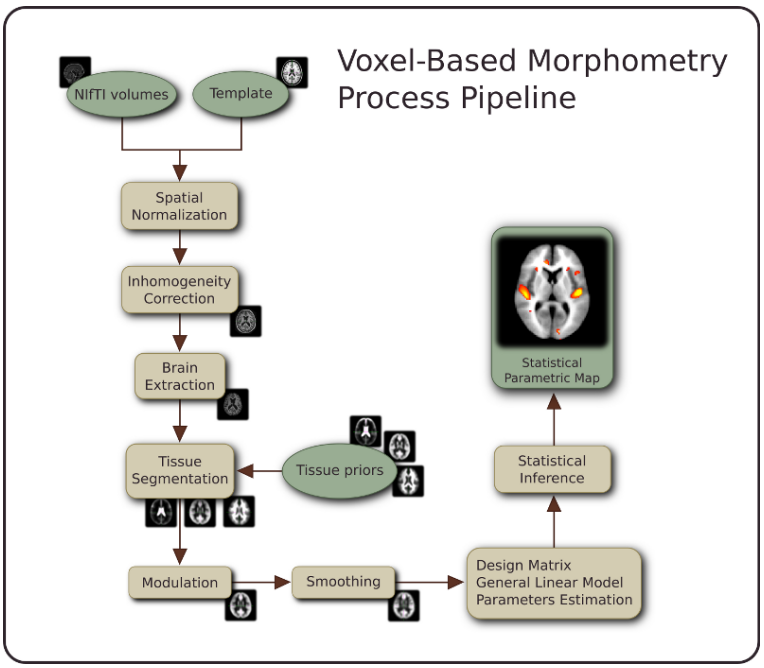


Figure A.2: The processing pipeline of the Voxel Based Morphometry (VBM) on structural MRI volumes.

A.4 OASIS anatomical imaging feature dataset

For specific classification experiments we have used a feature dataset which has been computed from a selected collection subjects from the Open Access Series of Imaging Studies (OASIS) database [**?**]. OASIS data set has a cross-sectional collection of 416 subjects covering the adult life span aged 18 to 96 including individuals with early-stage Alzheimer's Disease. The research group has used this particular database in previous works [**?**, **?**, **?**]

In this section we give a description of the feature extraction process that has been performed for the realization of the computational experiments on the classification of structural MRI data into Alzheimer's Disease (AD) patients and controls. The data sets are published in the research group web page [1] to allow the independent validation of our results, and to allow new classification approaches to be tested on this data. The process consists in the realization of Voxel Based Morphometry analysis to detect clusters of voxels which can be discriminat about AD patients and controls. These clusters are used as an spatial mask to extract from the original volumes the values of the corresponding voxels. We compute the mean and standard deviation of the voxel values inside each of the clusters, constituting the MSD feature vectors.

Selected Subjects from the OASIS database Many of the classification studies on the detection of AD were done with both men and women. However, it has been demonstrated that brains of women are different from men's to the extent that it is possible to discriminate the gender via MRI analysis. Moreover, it has been shown that VBM is sensitive to the gender differences. For these reasons, we have been very cautious in this study. We have selected from the OASIS a set of 98 MRI women's (aged 65-96 yr) T1-weighted MRI brain volumes. It must be noted that this is a large number of subjects compared with the other studies at the time of publication of our works. We have ruled out a set of 200 subjects whose demographic, clinical or derived anatomic volumes information was incomplete. For the present study there are 49 subjects who have been diagnosed with very mild to mild AD and 49 non-demented. A summary of subject demographics and dementia status is shown in table A.4.

[1] `http://www.ehu.es/ccwintco/index.php/GIC-experimental-databases`

	Very mild to mild AD	CS
No. of subjects	49	49
Age	78.08 (66-96)	77.77 (65-94)
Education	2.63 (1-5)	2.87 (1-5)
Socioeconomic status	2.94 (1-5)	2.88 (1-5)
CDR (0.5 / 1 / 2)	31 / 17 / 1	0
MMSE	24 (15-30)	28.96 (26-30)

Table A.1: Summary of subject demographics and dementia status. Education codes correspond to the following levels of education: 1-less than high school grad., 2-high school grad., 3-some college, 4-college grad., 5- beyond college. Categories of socioeconomic status: from 1 (biggest status) to 5 (lowest status). Clinical Dementia Rating (CDR). Mini-Mental State Examination (MMSE) score ranges from 0 (worst) to 30 (best).

Anatomical Imaging Protocol The OASIS database has been built following a strict imaging protocol, to avoid variations due to imaging protocol which would pose big image normalization problems. Multiple (three or four) high-resolution structural T1-weighted magnetization-prepared rapid gradient echo (MP-RAGE) images were acquired [**?**] on a 1.5-T Vision scanner (Siemens, Erlangen, Germany) in a single imaging session. Image parameters: TR= 9.7 msec., TE= 4.0 msec., Flip angle= 10, TI= 20 msec., TD= 200 msec., 128 sagittal 1.25 mm slices without gaps and pixels resolution of 256×256 (1×1mm).

Image processing and VBM for OASIS We have used the average MRI volume for each subject, provided in the OASIS data set. These images are already registered and re-sampled into a 1-mm isotropic image in atlas space and the bias field has been already corrected [**?**]. The Statistical Parametric Mapping software (SPM8) [**?**] was used to compute the VBM which gives us the spatial mask to obtain the classification features. Images were reoriented into a right-handed coordinate system to work with SPM8. The tissue segmentation step does not need to perform bias correction. We performed the modulation normalization for GM, because we are interested in this tissue for this study. We performed a spatial smoothing before performing the voxel-wise statistics, setting the FWHM of the Gaussian kernel to 10mm isotropic. A GM mask was created from the average of the GM segmentation volumes of the subjects under study. Thresholding the average GM segmentation, we obtain a binary mask that includes all voxels with probability

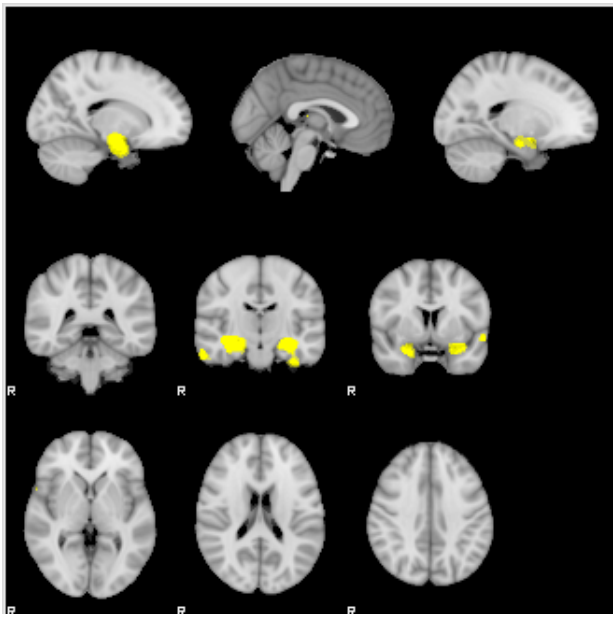


Figure A.3: SPM results: clusters of significant voxels with increased gray matter density in the controls relative to the patient subjects, detected by the VBM process.

greater than 0.1 in the average GM segmentation volume. This interpretation is not completely true, since the data is modulated, but it is close enough for the mask to be reasonable. We designed the statistical analysis as a Two-sample t-test in which the first group corresponds with AD subjects. In SPM software jargon: the contrast has been set to [-1 1], a right-tailed (groupN > groupAD), correction FWE, p-value=0.05. The VBM detected clusters are used for the feature extraction for the classification procedures. Statistical significance was determined using an extent threshold of 0 adjacent voxels for two sample comparisons. The clusters of significant voxels detected by the VBM analysis are displayed in figure A.3.

Computing the feature vectors The feature vector extraction processes is based on the voxel location clusters detected as a result of a VBM analysis . The VBM detected clusters are used as masks to determine the voxel positions where the features are extracted. These masks are applied to the GM density volumes result of the segmentation step in the VBM analysis. The feature extraction process computes the mean and standard deviation of the GM voxel values of each voxel location cluster.

A.5 resting state fMRI for Schizophrenia

Functional disconnection between brain regions is suspected to cause Schizophrenia symptoms, because of known aberrant effects on gray and white matter in brain regions that overlap with the default mode network. Resting state fMRI studies [**?**, **?**, **?**] have indicated aberrant default mode functional connectivity in schizophrenic patients. These studies suggest an important role for the default mode network in the pathophysiology of schizophrenia. Functional disconnectivity in schizophrenia could be expressed in altered connectivity of specific functional connections and/or functional networks, but it could also be related to a changed organization of the functional brain network. Resting state studies for schizophrenia patients with auditory hallucinations have also been performed [**?**] showing reduced connectivity. Recent findings [**?**] show effects on the resting state network localizations correlated with voxels in the left Heschl's gyrus (LHG; MNI coordinates -42,-26,10) from the auditory cortex effect related to the auditory hallucinations in schizophrenic patients.

Materials The results shown in this section are explorations over resting state fMRI data obtained from a 28 healthy control subjects (NC), and two groups of schizophrenia patients: 26 subjects with and 14 subjects without auditory hallucinations (SZAH and SZnAH respectively), selected from an on-going study in the McLean Hospital, Boston, Ma. Details of image acquisition and demographic information will be given elsewhere. For each subject we have 240 BOLD volumes and one T1-weighted anatomical image.

Preprocessing pipeline

The data preprocessing begins with the skull extraction using the Brain Extraction Tool from FSL [2]. All the images were manually oriented to AC-PC line. The functional images were coregistered to the T1-weighted anatomical image. Further preprocessing, including slice timing, head motion correction (a least squares approach and a 6-parameter spatial transformation), smoothing (FWHM=4mm) and spatial normalization to the Montreal Neurological Institute (MNI) template (resampling voxel size = 3 mm × 3 mm × 3 mm), temporal filtering (0.01-0.08 Hz) and linear trend removing, were conducted using the DPARSF [3] package. All the

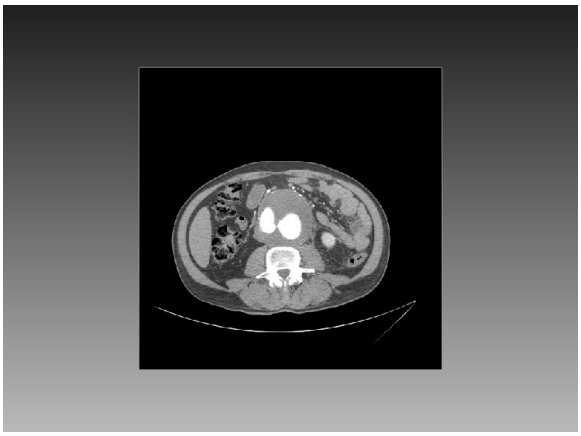
[2] `http://www.fmrib.ox.ac.uk/fsl/`

[3] `http://www.restfmri.net/forum/DPARSF`

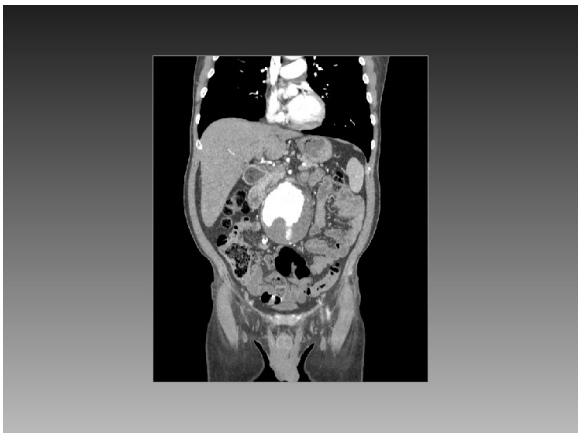
subjects have less than 3mm maximum displacement and less than 3º of angular motion.

A.6 Abdominal Aortic Aneurysm data Datasets

The data was provided by Dr Joszkowicz from the Mount Sinai School of Medicine (New York). Patients were administered 100cc of non-iodinated contrast agent with a rapid injection aid at 3-4cc per sec. The CTAs consist of 512×512 pixels per slice with physical voxel sizes in the 0.7- 1.2mm range. The datasets included various sizes and locations of the thrombus. Some of them were acquired after stent placement, and thus include strong streaking artifacts. The datasets were acquired on a 64-row CT scanner (Brilliance 64 - Phillips Healthcare, Cleveland, OH) and were chosen randomly from the hospital archive to represent wide variety of patients with different ages. A.4(a) and A.4(b) show an axial view and a coronal view of this dataset.



(a)



(b)

Figure A.4: Mount Sinai CT image with contrast agent. (a) Axial view (b) Coronal view

Bibliography

Bibliography