A Lattice Computing Approach to Alzheimer's Disease Computer Assisted Diagnosis Based on MRI Data

G.A. Papakostas[1], A. Savio[2], M. Graña[2], V.G. Kaburlasos[1]

[1]*Human Machines Interaction (HMI) Laboratory, Department of Computer and Informatics Engineering, TEI of Eastern Macedonia & Thrace,*
GR-65404 Agios Loukas, Kavala, Greece
e-mail: gpapak@teikav.edu.gr, vgkabs@teikav.edu.gr

[2]*Computational Intelligence Group, UPV/EHU, Spain*
e-mail: alexsavio@gmail.com, ccpgrrom@gmail.com

Abstract

We present a Computer Assisted Diagnosis (CAD) system for Alzheimer's disease (AD). The proposed CAD system employs MRI data features by applying a Lattice Computing (LC) approach. To this end feature extraction methods are adopted from the literature, which provide information allowing the discrimination of healthy people from Alzheimer diseased ones. The detection process is accomplished by a k-NN classifier defined in the LC context by handling this task from two different perspectives. Firstly, it performs dimensionality reduction over the high dimensional feature vectors and, secondly it classifies the subjects inside the lattice space by generating adaptively class boundaries. Computational experiments using a benchmark dataset regarding MRI data on AD patients have shown that the proposed classifier performs competitively to state-of-the-art classification models.

Keywords: Alzheimer's disease, lattice computing, MRI, classification.

1. Introduction

Alzheimer's disease (AD) is becoming a form of pandemia all over the world (with a remarkable increase in developing countries) affecting people over 65 years old [1]. Currently there is no cure for this disease. However, an early non-invasive detection of AD can improve the life quality of the patients and their families. It can also help researchers in the development of appropriate treatments. Moreover, due to the significant economical impact of AD to society [2], the development of a system for an early automatic AD detection is economically highly desirable.

Much effort has been expended in this direction, resulting in a variety of approaches for early AD detection using MRI data. These approaches address the extraction of discriminative features [3-8], the selection of the most appropriate features [6,9] from a large pool, and the selection of efficient classification models [5,10,11] from machine learning and computational intelligence. This paper reports works in the last two research directions, namely the feature vector dimensional reduction and the development of an efficient classifier.

Both aforementioned two objectives are pursued here in the framework of Lattice Computing (LC). Lattice Computing was initially defined as "the collection of Computational Intelligence tools and techniques that either make use of lattice operators *inf (infimum)* and *sup (supremum)* for the construction of the computational algorithms or exploit Lattice Theory for language representation and reasoning" [12]. Lattice computing techniques have been used successfully in a number of applications including, industrial dispensing [13], structure identification [14], human facial expression recognition [15], face recognition using thermal infrared images [16], etc.

The contribution of this work is twofold: (1) the employment of the LC framework for dimensionality reduction of the feature vectors extracted from Magnetic Resonance Imaging (MRI) data, and (2) the detection of Alzheimer's disease. The former novelty is accomplished by the Intervals' Numbers (INs) meta-representation enabling the transformation of the feature vectors to *h-cuts* of user defined size. The extracted *h-cuts* vectors are used to classify subjects into healthy and AD classes by an evolutionary adjustable k-NN classifier defined in the LC context.

The reported experimental results on the OASIS [17] benchmark MRI dataset are promising. The proposed classifier has demonstrated comparable results with some state-of-the-art classification models (e.g. neural networks, SVM), in terms of features vectors' dimension, classification accuracy, sensitivity and specificity.

The paper is organized as follows. Section 2 details the feature extraction methods applied to describe the MRI data. Section 3 discusses the properties of the proposed minimum distance classifier defined in the lattice space. Section 4 presents experimental results, comparing alternative classification models, as well as alternative feature extraction techniques from the literature. Finally, section 5 concludes by summarizing our contribution and discussing future work.

2. Feature Extraction

We have considered two different feature extraction methods, resulting in two different datasets for the computational experiments. More specifically, the first approach uses the results of a Voxel-based Morphometry (VBM) analysis, whereas the second approach uses scalar maps computed from the deformation based morphometry (DBM) analysis, as detailed next.

2.1 Voxel-based Morphometry

Morphometry analysis is a widely used tool in clinical research for brain anatomy studies based on MRI. It allows a comprehensive measurement of anatomical differences within a group or across groups throughout the entire brain.

Specifically, VBM is an approach that measures differences in local concentrations of brain tissue. Often grey matter is considered in the studies through a voxel-wise comparison of brain MRI volumes (in the form of NIfTI “Neuroimaging Informatics Technology Initiative” data files) across subjects. The result of a VBM analysis is a thresholded statistical parametric map (SPM) which is used here as a spatial mask specifying the voxels sites selected to obtain a feature vector from each subject. These feature vectors (denoted MSD in the results section) are composed of the mean and standard deviation of each voxel cluster in the SPM. The processing pipeline of VBM is illustrated in Fig. 1, while the overall procedure is explained in more detail in [5].

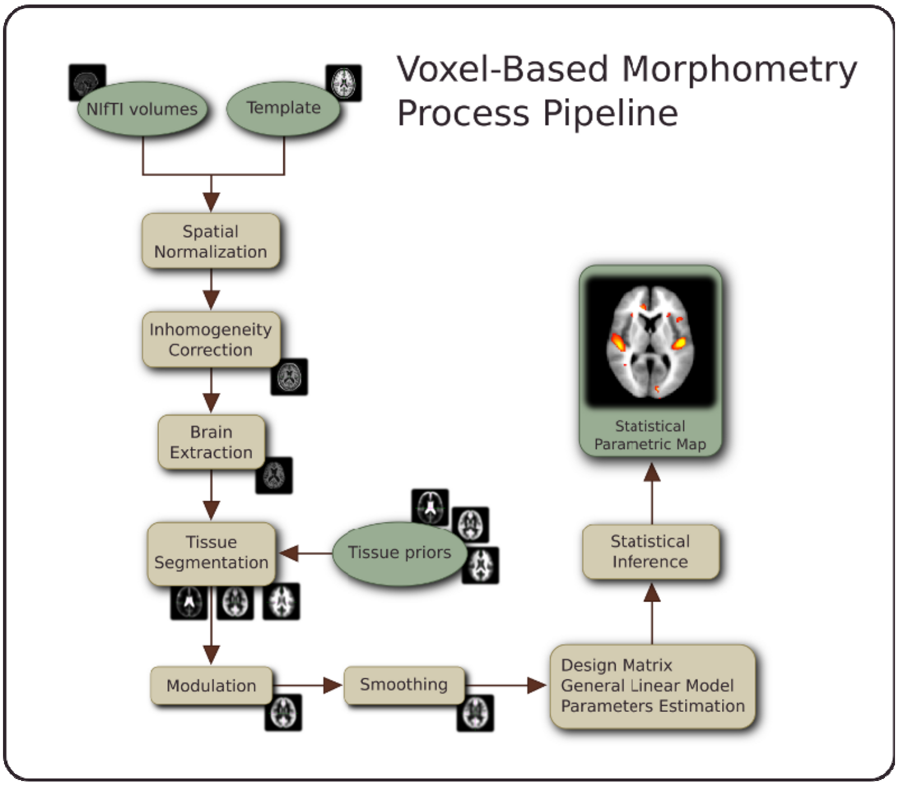

Figure 1: Pipeline of the VBM procedure.

2.2 Deformation-based Morphometry

Deformation-based Morphometry (DBM) uses non-linear spatial normalization data to find differences between groups. We built a custom brain template with the subjects in OASIS, performing a non-linear registration [18] of selected 98 subjects. As a result, one volume of 3D displacement vectors for each subject is obtained. These displacement vector fields provide the details of the deformation of the template brain to the subject's MRI data. For each voxel *i*, the displacement field for one subject has a vector (x_i, y_i, z_i) representing the ending point, in millimetres, of voxel *i* in the registration process. We compute two scalar maps from these displacement vectors [4]: the displacement magnitude (DM) and the Jacobian determinant of the field gradient matrices (JD). The processing pipeline for feature extraction from DBM procedure is illustrated in Fig. 2.

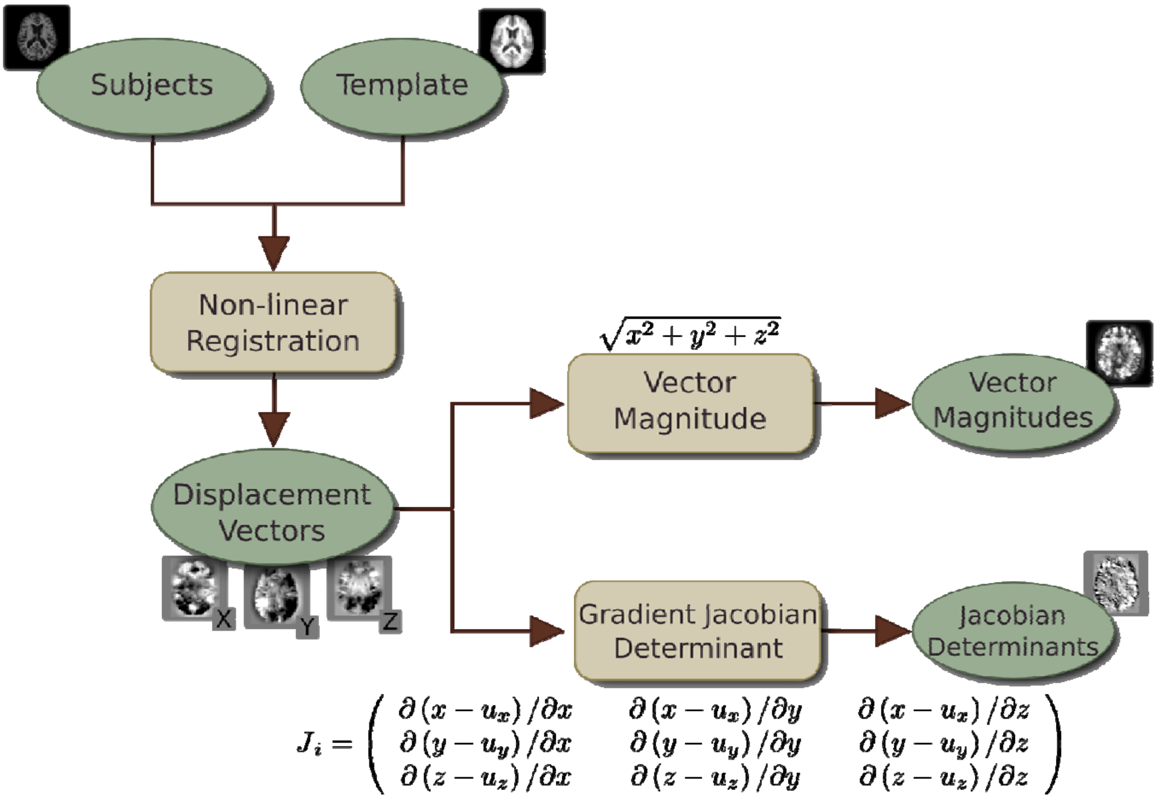
Subjects
Template
Non-linear Registration
Displacement Vectors
X
Y
Z
$\sqrt{x^2+y^2+z^2}$
Vector Magnitude
Vector Magnitudes
Gradient Jacobian Determinant
Jacobian Determinants
$J_i = \begin{pmatrix} \partial(x-u_x)/\partial x & \partial(x-u_x)/\partial y & \partial(x-u_x)/\partial z \\ \partial(y-u_y)/\partial x & \partial(y-u_y)/\partial y & \partial(y-u_y)/\partial z \\ \partial(z-u_z)/\partial x & \partial(z-u_z)/\partial y & \partial(z-u_z)/\partial z \end{pmatrix}$

Figure 2: Pipeline of the DBM procedure.

Note that the pipeline of the DBM procedure of Fig.2 starts from noise corrected and skull-stripped volumes. Feature selection is accomplished by the computation of the correlation between the scalar maps across subjects and the categorical variable (0 healthy controls, 1 AD patients) for each voxel independently. We compute the empirical distribution of the absolute (Pearson, Spearman) correlation values, we select voxel sites according to a set percentile on this distribution. Specifically, in the experiments we use 0.990, 0.995 and 0.999 percentiles.

3. Classification in the Lattice Computing (LC) context

This section delineates the mathematical background, namely *lattice theory*, and it presents novel classification tools.

3.1 Basics

A lattice is a partially ordered set ($\mathbb{L}$,$\sqsubseteq$) with the additional property that any two elements $x, y \in L$ have both a greatest lower bound, namely *infimum*, denoted by $x \wedge y$ and a least upper bound, namely *supremum*, denoted by $x \vee y$. If in a lattice ($\mathbb{L}$,$\sqsubseteq$) every (x, y) pair satisfies either x⊑y or x⊒y then we say that the lattice ($\mathbb{L}$,⊑) is totally-ordered. Moreover, a lattice ($\mathbb{L}$,⊑)is called *complete* iff each of its subsets X has both an infimum and a supremum in $\mathbb{L}$ (hence, taking $X = \mathbb{L}$, we see that a complete lattice has both a least element and a greatest element) [19-20].

Based on the above definitions, novel data-mining and classification techniques have been proposed lately, into the *lattice space*, to deal successfully with disparate types of numerical and nominal data [21]. By *lattice space* we mean a set of lattice-ordered data. Our interest here is in developing a classification scheme in the lattice space of Interval's Numbers [22] defined next.

3.2 INs Meta-Representation

In the first place, we define a (Type-1) interval.

Definition 3.1. A (Type-1) *interval* $A=[\alpha_1,\alpha_2]$ is defined by

$[\alpha_1,\alpha_2]=\{x : x\in\mathbb{R} \ and \ \alpha_1 \le x \le \alpha_2\}$.

Let $\mathbb{I}$ denote the set of intervals on the real line including also the empty interval (is identical to the empty set).

A (Type-1) Intervals' Number is defined next:

Definition 3.2. An Intervals' Number (IN) is a function $\mathbb{F}$: [0,1]→$\mathbb{I}$ which satisfies

$$\begin{aligned}&1)\, h_1 \ge h_2 \Rightarrow F_{h_1} \subseteq F_{h_2}.\\&2)\, \forall P\subseteq[0,1]:\bigcap_{h\in P} F_h = F_{\vee P}.\end{aligned} \qquad (1)$$

where $F_h \in \mathbb{I}$ denotes the image (interval) of $h\in[0,1]$ and $\forall P$ the supremum of the set $P\subseteq[0,1]$. We remark that the "$\subseteq$" in Definition 3.2 denotes the conventional set-inclusion.

Based on the following theorem one can associate every IN to a fuzzy interval [23].

We define $\mathbb{F}_1$ the set of all (Type-1) INs.

Theorem 3.3. Given IN $E\in\mathbb{F}_1$, a fuzzy set $\tilde{E}$ as $\forall x:\tilde{E}(x)=\sup\{h : x\in E_h\}$ is defined.

where E_h is the *h-cut* (identical to the *α-cut* of *fuzzy theory*) of IN E. The *h-cuts* of $\tilde{E}$ are denoted by $\tilde{E}_h$ and by definition, satisfy: $\forall h\in[0,1]:\tilde{E}_h=\{x:\tilde{E}(x)\ge h\}$. Then, for all $h\in[0,1]$ we have $\tilde{E}_h=\tilde{E}(x)$. Hence, E is a *fuzzy interval*.

Based on the above, there follow two equivalent representations for a Type-1 IN, namely the *interval-representation* and the *membership-function-representation* [22] illustrated below.

Furthermore, given an IN F the symbol $F_h \in \mathbb{I}$, $h \in [0,1]$ is attached to its interval-representation, whereas the symbol $F(x) \in [0,1], x \in \mathbb{R}$ is attached to its membership-function representation.

It is already known [24] that any set of fuzzy intervals is equipped with the fuzzy sets order relation $\leq$, is a complete lattice. Just like fuzzy intervals, INs are equipped with a partial order relation $\preceq$ as follows:

$$F \preceq G \Leftrightarrow \left(\forall h \in [0,1] : F_h \subseteq G_h\right) \tag{2}$$

with $F, G \in \mathbb{F}_1$. The relation $\preceq$ is a lattice order and the lattice $(\mathbb{F}_1, \preceq)$ of INs is complete.

3.2.1 INs Extraction

Considering a population of data samples (features), an algorithm [13, 16] for inducing an IN needs to be applied according to the following description.

Suppose a set X of real data samples, e.g. X $=(x_1,\ldots,x_n)$. Two elements x_i, x_j of the set X are called *successive* iff there is no other element $x_k, k \in \{1,\ldots,n\}$ such that $x_i \wedge x_j < x_k < x_i \vee x_j$, where $\wedge$ and $\vee$ are the min and max operators, respectively. A strictly-increasing, cumulative real function $c:\Re \rightarrow \Re_0^+$ is computed by defining

$$c(x_i) = \frac{1}{n}\left|\left\{x_j : j \in \{1,\ldots,n\} \text{ and } x_j \leq x_i\right\}\right|, \; i \in \{1,\ldots,n\} \tag{3}$$

where $|S|$ denotes the cardinality of the set S; finally, function $c:\Re \rightarrow \Re_0^+$ is defined by straight-line connecting any two points $(x_i, c(x_i))$ and $(x_j, c(x_j))$, where x_i, x_j are successive elements of set X. Obviously, there is a unique real number x_0 such that $c(x_0) = 0.5$. In conclusion, an IN is calculated from function $c(\cdot)$ such as for values less-than or equal-to x_0 the corresponding IN envelope function is $2c(x)$, whereas for values larger than x_0 the corresponding IN envelope function is $2(1-c(x))$. An example of a Type-1 IN induction from a data samples population is detailed in Fig.3.

Furthermore, an IN envelope is represented by a user-defined number of N_h equally spaced intervals from $h=0$ to $h=1$ (Fig.3d) and thus any population of data samples in the set X can be represented by N_h intervals stored in a N_hx2 size matrix.

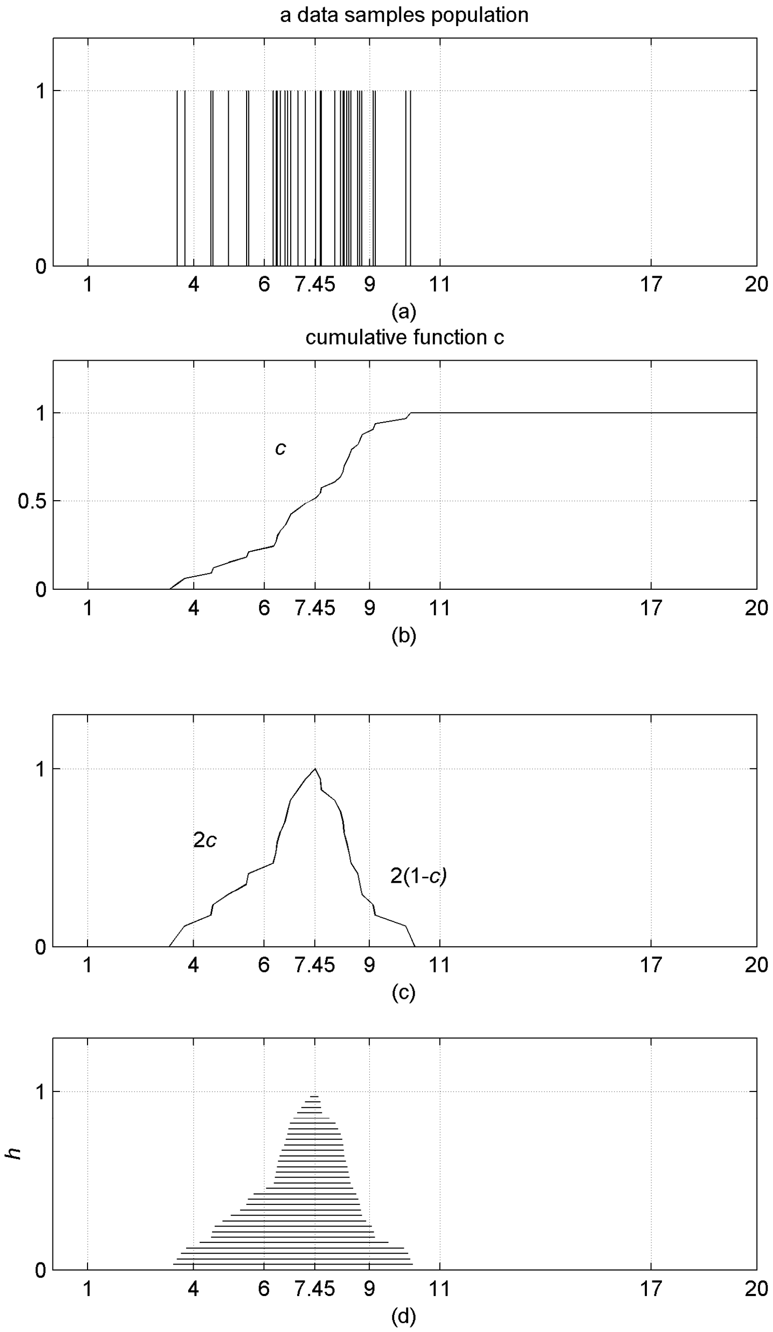

Figure 3: Induction of a Type-1 IN from a data samples population whose median value equals 7.45. (a) the data samples population, (b) the cumulative function c, (c) the membership-function representation of a Type-1 IN and (d) the corresponding intervals representation a Type-1 IN (N_h =32).

The representation of a distribution of data samples by an IN enables the employment of established, useful mathematical instruments for measuring a distance as well as for calculating a (fuzzy) degree of partial order of different data distributions [15, 22].

3.3 Dimensionality Reduction with INs

It is worth noting that the IN meta-representation includes all order data statistics of the initial population. This characteristic along with the ability to provide an IN meta-representation of any size for the data set X, makes this meta-representation appealing for big (numeric) data, such as in Alzheimer's disease detection, where initial feature vector sizes are of the order of several hundreds or even thousands..

Dimensionality reduction is achieved by ad-hoc selecting the number N_h of *h-cuts* such that $N_h \ll (DataSize/2)$ where *DataSize* is the size of the initial population data samples. Although, a number of N_h =32 *h-cuts* is usually used [15,22], several numbers up to *DataSize* can be examined.

3.4 Lattice Computing k-NN (LC-kNN)

An extension of the well known k-NN classifier in the LC context is described in this section. The main functionality of the k-NN classifier is the computation of the distances between the training samples and a test sample to find the nearest neighbours. Finally, the sample is classified according to a majority vote on the classes of its *k* nearest neighbours.

The k-NN classifier can be reformulated in the LC framework by incorporating a distance (metric) over the meta-representation of the samples as INs. Such a distance is defined as follows:

Definition 3.4. A metric distance [25] *d* in a nonempty set *A* is a real function d:A×A→ [0,+∞) which satisfies the following conditions, $\forall\, x, y, z \in A$:

C1. $d(x,y)=0 \Leftrightarrow x=y$ (coincidence)

C2. $d(x,y)=d(y,x)$ (symmetry)

C3. $d(x,z)+d(z,y)\geq d(x,y)$ (triangle inequality)

A distance function $d_{\mathbb{F}}:\mathbb{F}\times\mathbb{F}\rightarrow\mathfrak{R}_0^+$ in the lattice $(\mathbb{F},\preceq)$ of INs is defined [16] as:

$$d_{\mathrm{F}}(F,G)=\int_0^1 d_{\mathrm{I}}(F_h,G_h)\,dh \tag{4}$$

where function $d_{\mathrm{I}}:\mathrm{I}\times\mathrm{I}\to\Re_0^+$ is a distance in the lattice $(\mathrm{I},\preceq)$ given by

$$\begin{aligned} d_{\mathrm{I}}([a,b],[c,d])=&\left[v(\theta(a\wedge c))-v(\theta(a\vee c))\right]\\ &+\left[v(b\vee d)-v(b\wedge d)\right]; \end{aligned} \tag{5}$$

where $v:\Re\to\Re_0^+$ and $\theta:\Re\to\Re$ are strictly increasing and decreasing functions, respectively.

In conclusion, a distance function $d:\mathbb{F}^N\times\mathbb{F}^N\to\Re_0^+$ is given by

$$d(\mathbf{F},\mathbf{G})=d((F_1,\ldots,F_N),(G_1,\ldots,G_N))=\sum_{i=1}^{N}d_{\mathrm{F}}(F_i,G_i) \tag{6}$$

where **F,G** are N-tuples of INs, i.e. $\mathbf{F}=(F_1,\ldots,F_N)$.

The following functions $v:\Re\to\Re_0^+$ and $\theta:\Re\to\Re$ have been used in Eq.(5)

$$v(x)=\frac{A}{1+e^{-\lambda(x-\mu)}} \quad \text{and} \quad \theta(x)=2\mu-x \tag{7}$$

where $A,\lambda\in\Re^+$ and $\mu\in\Re$ are free parameters increasing the flexibility of the distance function when comparing two INs at different resolutions which need to be tuned appropriately. The estimation of the three parameters of Eq.(7) can be performed by stochastic (e.g., genetic algorithm, PSO, ACO) optimization techniques.

4. Experimental Study

In order to investigate the detection performance of the proposed LC-kNN classifier, a set of appropriate experiments were conducted. For the experimental purposes, specific software was developed in MATLAB 2012b integrated development environment. All experiments were executed in an Intel i5 3.3GHz PC with 8GB RAM.

4.1 Dataset

A subset of the Open Access Series of Imaging Studies (OASIS) database [17] was selected in order to evaluate the detection performance of the proposed LC-kNN classifier. This two-class dataset has a cross-sectional collection of 416 subjects covering the adult life span aged 18–96 including individuals with early-stage Alzheimer's disease. A subset of the original OASIS dataset including 98 right-handed women (aged 65-96 yr) is considered herein. More precisely, the used subset consists of 49 subjects who have been diagnosed with very mild to mild AD (class 1) and 49 non-demented (class 2). A summary of subject demographics and dementia status is shown in [5]. The feature extraction methods (MSD, DM, JD) discussed in section 2 were adopted to build the datasets used in the experimental evaluation of an AD classification system using on the lattice based LC-kNN.

4.2 Experimental Results

A “10-fold cross-validation” validation strategy was applied, hence the dataset is partitioned in ten parts with one-tenth of the data being used for testing, whereas the remaining nine-tenths are used for training the classifier. The experiment is repeated using a different partition as test in each turn, so that all tenths of the data were used for testing. Care was taken so that all classes are represented equally in both the training and the testing data.

The actual feature datasets have been used in several works in the literature, hence results obtained with a variety of classifier models are publicly available for comparison. The AD detection statistics, accuracy, sensitivity and specificity (mean values and standard deviation inside parentheses) for the OASIS subset of 98 subjects, of the proposed LC-kNN classifier in comparison with the state-of-the-art models [5, 11] using the MSD features are summarized in Table 1.

Table 1. AD detection statistics of several classifiers using the MSD features. Bold typeface identifies performances above 0.80.

Feature Type	Classifier Type	#Features	Accuracy	Sensitivity	Specificity
MSD	MLP-BP	24	0.78 (0.12)	0.69 (0.14)	0.88 (0.13)
	RBF		0.66 (0.13)	0.65 (0.24)	0.68 (0.14)
	PNN		0.78 (0.09)	0.62 (0.14)	0.94 (0.10)
	LVQI		**0.81 (0.18)**	**0.72 (0.27)**	**0.90 (0.14)**
	LVQ2		**0.83 (0.12)**	**0.74 (0.23)**	**0.92 (0.10)**
	Linear SVM		0.78 (N/A)	0.72 (N/A)	0.88 (N/A)
	rbf SVM		**0.81 (N/A)**	**0.75 (N/A)**	**0.89 (N/A)**
	Indep-linear-SVM		0.74 (N/A)	0.51 (N/A)	0.97 (N/A)
	Indep-rbf-SVM		0.75 (N/A)	0.56 (N/A)	0.95 (N/A)
	linear-AB-SVM		0.71 (N/A)	0.54 (N/A)	0.88 (N/A)
	rbf-AB-SVM		0.79 (N/A)	0.78 (N/A)	0.80 (N/A)
	rbf-DAB-SVM		**0.85 (N/A)**	**0.78 (N/A)**	**0.92 (N/A)**
	Kernel-LICA-DC		0.74 (N/A)	0.96 (N/A)	0.52 (N/A)
	LC-kNN (k=3)	12	**0.80 (0.13)**	**0.80 (0.19)**	**0.79 (0.20)**

By examining the results of Table 1, it is deduced that the LC-kNN with k=3 show superior detection performance than some conventional classifiers such as MLP-BP, RBF, PNN, Linear SVM and some advanced classification models (Indep-linear-SVM, Indep-rbf-SVM, linear-AB-SVM, rbf-AB-SVM, Kernel-LICA-DC). Moreover, it performs comparatively well with the most effective models LVQ1, LVQ2, rbf- SVM (lower by 1%-3%), while it is worst by 5% than the rbf-DAB-SVM classifier. However, it is worth noting that although the proposed LC-kNN has 5% lower detection rate, it uses 50% smaller feature vectors (12 instead of 24) than all the other classifiers under comparison.

Considering the DM (Displacement Magnitude - see section 2.2) feature type, the LC-kNN outperforms the linear SVM classifier [4] with extremely lower feature dimension. Moreover, the LC-kNN classifier retains its superiority even when a dimensionality reduction process [4] is applied to the other classifiers. The detection performance of the classifiers for the case of the DM features and Pearson (Pe), Spearman (Sp) correlation functions is presented in the following Table 2.

Table 2. AD detection statistics of several classifiers using the DM features. Bold typeface identifies performances above 0.80.

Feature Type	Classifier Type	#Features	Accuracy	Sensitivity	Specificity
DM_Sp995	Linear SVM	12229	0.76 (0.15)	0.77 (0.28)	0.75 (0.17)
		250	0.79 (0.10)	0.90 (0.13)	0.67 (0.17)
	LC-kNN (k=3)	64	**0.82 (0.09)**	**0.86 (0.14)**	**0.78 (0.22)**
DM_Sp999	Linear SVM	1861	0.66 (0.14)	0.70 (0.20)	0.62 (0.21)
		250	0.72 (0.15)	0.80 (0.11)	0.65 (0.29)
	LC-kNN (k=3)	64	**0.82 (0.13)**	**0.88 (0.14)**	**0.76 (0.21)**
DM_Pe995	**Linear SVM**	27474	**0.84 (0.10)**	**0.90 (0.17)**	**0.77 (0.14)**
		250	**0.84 (0.10)**	**0.92 (0.12)**	**0.75 (0.17)**
	LC-kNN (k=3)	64	**0.83 (0.04)**	**0.84 (0.13)**	**0.82 (0.18)**

The above results clearly show the superiority of the LC-kNN classifier against the popular Linear SVM, since it detects the AD with 3%-10% higher accuracy by using 75% less features. This performance is very impressive and highlights the ability of the *h-cuts* and therefore of the INs meta-representation, to embody discriminant information for accurate classification.

Following the same experimental protocol, the third JD feature type was applied and the corresponding detection statistics are summarized in Table 3.

The results for the case of JD (Jacobian Determinant - see section 2.2) features and Pearson (Pe), Spearman (Sp) correlation functions are similar with the previous ones. More precisely, the LC-kNN classifier shows 3% higher accuracy or performs equal than the Linear SVM model in the worst case. However, the dimension of the feature vector used by LC-kNN is extremely lower (30 times) than that of SVM, 64 instead of 2000.

Table 3. AD detection statistics of several classifiers using the JD features. Bold typeface identifies the highest performances.

Feature Type	Classifier Type	#Features	Accuracy	Sensitivity	Specificity
JD_Sp995	Linear - SVM	17982	0.76 (0.14)	0.77 (0.27)	0.75 (0.16)
		2000	0.65 (0.15)	0.65 (0.21)	0.65 (0.24)
	LC-kNN (k=3)	64	**0.79 (0.11)**	**0.74 (0.21)**	**0.84 (0.16)**
JD_Pe990	**Linear - SVM**	43967	0.66 (0.19)	0.70 (0.20)	0.62 (0.24)
		2000	**0.75 (0.13)**	**0.75 (0.26)**	**0.75 (0.17)**
	LC-kNN (k=3)	64	0.74 (0.11)	0.82 (0.15)	0.66 (0.21)

The reduction of the data dimensions is achieved by selecting the N_h parameter that controls the number of extracted *h-cuts*. This selection of the control parameter (N_h) is performed empirically (by trial and error) without taking into account specific data-driven information. Therefore, it is not a priori known what is the optimum value of N_h. Moreover, due to the fact that the INs representation is a meta-representation of the initial data vectors, these two information carriers have different accuracy independent on their size. Thus the lower dimensionality reduction does not guarantee a higher accuracy.

However the authors have scheduled a future research in investigating that reduces the dimension (selection of N_h parameter) of the data vectors by considering the classes' distributions and separability.

The aforementioned experimental analysis demonstrates the potential of the lattice computing framework through the remarkable high performance of the proposed LC-kNN classifier in terms of accuracy and representation simplicity.

5. Conclusion

A new k-NN classifier defined in the LC context was proposed. The introduced LC-kNN model was applied in AD disease detection showing promising results, since it detected the diseased subjects accurately enough by using features of high compactness (low dimension) and discriminability. The INs representation along with the adaptive nature of the used distance metric defining in the lattice space, constitute the main factors of the LC-kNN classifier efficiency. Although the results are acceptable and in some cases impressive, some additional actions such as the application of the LC-kNN to the entire OASIS dataset and the employment of other feature types, need to be scheduled for future work.

References

[1] Alzheimer's Association, http://www.alz.org/.

[2] S. Bonin-Guillaume, D. Zekry, E. Giacobini, G. Gold, JP. Michel, "Impact économique de la démence (English: The Economical Impact of Dementia)", *La Presse Médicale*, vol. 34, no. 1, pp. 35–41, 2005.

[3] E. Gerardin, G. Chetelat, M. Chupin, R. Cuingnet, B. Desgranges, Ho-Sung Kim, M. Niethammer, B. Dubois, S. Lehericy, L. Garnero, F. Eustache, and O. Colliot, "Multidimensional classification of hippocampal shape features discriminates Alzheimer's disease and mild cognitive impairment from normal aging", *NeuroImage*, vol. 47, no. 4, pp. 1476-1486, 2009.

[4] A. Savio, M. Graña, J. Villanúa, "Deformation based features for Alzheimer's disease detection with linear SVM", *Proceedings of the 6th international conference on Hybrid artificial intelligent systems (HAIS'11)*, vol. II, pp. 336-343, 2011.

[5] A. Savio, M.T. García-Sebastián, D. Chyzyk, C. Hernandez, M. Graña, A. Sistiaga, A. López de Munain, J. Villanúa, "Neurocognitive disorder detection based on feature vectors extracted from VBM analysis of structural MRI", *Computers in Biology and Medicine*, vol. 41, no. 8, pp. 600-610, 2011.

[6] A. Savio, M. Graña, "Deformation based feature selection for Computer Aided Diagnosis of Alzheimer's Disease", *Expert Systems with Applications*, vol. 40, no. 5, pp. 1619-1628, 2013.

[7] B. Ayerdi, A. Savio, M. Graña, "Meta-ensembles of classifiers for Alzheimer's disease detection using independent ROI features", 5th International Work-

Conference on the Interplay Between Natural and Artificial Computation, Part II, pp 122-130, Mallorca, Spain, June 10-14, 2013.

[8] A.-L. Fouque, C. Fischer, V. Frouin, P. Ciuciu, E. Duchesnay, "Comparison of features for voxel-based analysis and classification of anatomical neuroimaging data," *International Workshop on Pattern Recognition in Neuroimaging (PRNI),* pp. 186-189, 22-24 June 2013.

[9] R. Chaves, J. Ramírez, J.M. Górriz, C.G. Puntonet, "Association rule-based feature selection method for Alzheimer's disease diagnosis", *Expert Systems with Applications*, vol. 39, no. 14, pp. 11766-11774, 2012.

[10] C. Hernández, M. Graña, J. Villanúa, A. Savio, M. García-Sebastián, "Classification results of artificial neural networks for Alzheimer's disease detection", Intelligent Data Engineering and Automated Learning- IDEAL 2009, Emilio Corchado, HujunYin (eds) LNCS 5788, pages 641-648, 2009.

[11] D. Chyzhyk, M. Graña, A. Savio, J. Maiora, "Hybrid dendritic computing with kernel-LICA applied to Alzheimer's disease detection in MRI", *Neurocomputing*, vol. 75, no. 1, pp. 72-77, 2012.

[12] M. Graña, Ed., "Special issue on: Lattice computing and natural computing", *Neurocomputing*, vol. 72, no. 10-12, pp. 2065-2066, 2009.

[13] V. G. Kaburlasos and T. Pachidis, "A lattice-computing ensemble for reasoning based on formal fusion of disparate data types, and an industrial dispensing application", *Information Fusion*, vol. 16, pp. 68-83, 2014.

[14] V. G. Kaburlasos and S. E. Papadakis, "Granular self-organizing map (grSOM) for structure identification", *Neural Networks*, vol. 19, no. 5, pp. 623-643, 2006.

[15] V.G. Kaburlasos, S.E. Papadakis and G.A. Papakostas, "A Lattice Computing Extension of the FAM Neural Classifier for Human Facial Expression Recognition", *IEEE Transactions on Neural Networks & Learning Systems*, vol. 24, no. 10, pp. 1526 - 1538, 2013.

[16] G.A. Papakostas, V.G. Kaburlasos and Th. Pachidis, "Thermal Infrared Face Recognition Based on Lattice Computing (LC) Techniques", *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013)*, pp. 1-6, 7-10 July, Hyderabad - India, 2013,

[17] D.S. Marcus, T.H. Wang, J. Parker, J.G. Csernansky, J.C. Morris, and R.L. Buckner, "Open access series of imaging studies (OASIS): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults", *Journal of Cognitive Neuroscience*, vol. 19, no. 9, pp. 1498-1507, 2007.

[18] B.B. Avants, C.L. Epstein, M. Grossman, J.C. Gee, “Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain”, *Medical Image Analysis*, vol. 12, no. 1, pp. 26-41, 2008.

[19] G. Birkhoff, *Lattice Theory*. Providence, RI: American Mathematical Society, vol. 25, 3rd Edition, Colloquium Publications, 1967.

[20] V. G. Kaburlasos, *Towards a Unified Modeling and Knowledge-Representation Based on Lattice Theory*. Heidelberg, Germany: Springer, 2006, ser. Studies in Computational Intelligence, vol. 27.

[21] V.G. Kaburlasos and L. Moussiades, “Induction of formal concepts by lattice computing techniques toward tunable classification”, *Journal of Engineering Science and Technology Review*, vol. 7, no. 1, pp. 1-7, 2014.

[22] V. G. Kaburlasos and A. Kehagias, “Fuzzy inference system (FIS) extensions based on lattice theory”, *IEEE Transactions on Fuzzy Systems*, in press, doi: 10.1109/TFUZZ.2013.2263807.

[23] V.G. Kaburlasos, G.A. Papakostas, Th. Pachidis and A. Athinellis, “Intervals' Numbers (INs) Interpolation/Extrapolation”, *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2013)*, pp. 1-8, 7-10 July, 2013, Hyderabad - India.

[24] Ath. Kehagias, “Some remarks on the lattice of fuzzy intervals”, *Information Sciences*, vol. 181, no. 10, pp. 1863-1873, 2011.

[25] G.A. Papakostas, A.G. Hatzimichailidis and V.G. Kaburlasos, “Distance and similarity measures between intuitionistic fuzzy sets: A comparative analysis from a pattern recognition point of view”, *Pattern Recognition Letters*, vol. 34, no. 14, pp. 1609-1622, 2013.