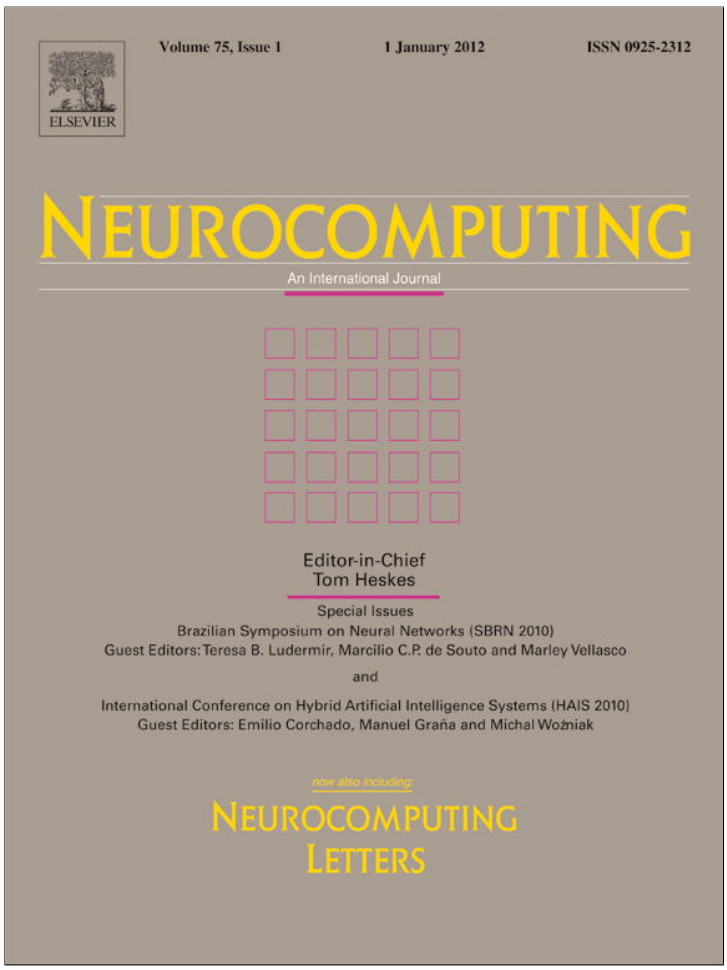

Neurocomputing 75 (2012) 72–77

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Hybrid dendritic computing with kernel-LICA applied to Alzheimer's disease detection in MRI

Darya Chyzhyk, Manuel Graña *, Alexandre Savio, Josu Maiora

Facultad de Informatica, Paseo Mauel Lardizabal 1, 20018 San Sebastian, Spain

ARTICLE INFO

Available online 4 August 2011

Keywords:
Dendritic computing
Lattice computing
Alzheimer's disease
Kernel method
Lattice Independent Component Analysis

ABSTRACT

Dendritic computing has been proved to produce perfect approximation of any data distribution. This result guarantees perfect accuracy training. However, we have found great performance degradation when tested on conventional *k*-fold cross-validation schemes. In this paper we propose to use Lattice Independent Component Analysis (LICA) and the Kernel transformation of the data as an appropriate feature extraction that improves the generalization of dendritic computing classifiers. 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.

1. Introduction

Dendritic computing (DC) [2,14,18,19,21] 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 [17,21]. However, it suffers from over-fitting problems. The results on cross-validation experiments result in very poor performance. We have confirmed that on a particular database that we have studied in previous works [4,22,23], we found that SNLDC showed high sensitivity but very low specificity in a 10-fold cross-validation experiment. These baseline results are reproduced below in Section 5. To improve the method, [2] proposed to compute the optimal rotation of each of the hyperboxes by some optimization method at each step of the training algorithm. This procedure is computationally very expensive and does not guarantee optimal generalization of classification performance. It depends on the local distribution of the data, as a local kernel transformation whose parameters must be fitted locally.

In this paper we propose to perform a transformation of the data which is appropriate for later DC based classification systems, following the trend of innovation through hybrid system design [1,3]. This transformation is composed of a kernel transformation [24] followed by dimension reduction process realized by Lattice Independent Component Analysis (LICA). The composite transformation is the Kernel-LICA approach. The kernel transformation is intended to produce a high dimensional feature representation of the data that eases the subsequent processes. The dimension reduction phase could be realized by other methods, such as Principal Component Analysis (PCA), which has also been tested in this paper. Notice that both dendritic computing and LICA are lattice computing [8] algorithms.

The target application of our work is the detection of Alzheimer's disease (AD) patients from brain magnetic resonance imaging (MRI) scans. We have worked over a database of MRI features[1] extracted from the OASIS database of MRI scans of AD patients and controls [23,22,4]. Specifically, we selected from the OASIS databases a subset of AD patients and controls of the same sex. This subsample from OASIS contains the same number of AD patients and controls, therefore is a well class balanced sample. Then we performed a Voxel Based Morphometry (VBM) analysis to determine the location of the voxel clusters most affected by the disease. These voxel clusters were collected in the gray matter segmentation of each MRI scan and used to compute feature vectors for classification. In this paper, the feature vectors are built gathering the mean and standard deviation of the voxel gray matter segmentation values of all and each of the detected clusters.

The structure of the paper is the following. Section 2 reviews the baseline dendritic approach used. Section 3 reviews the LICA approach. Section 4 describes our application of the kernel trick to LICA. Section 5 gives our experimental results on the AD database. Section 6 gives our conclusions.

* Corresponding author. Tel.: +34 943 018044; fax: +34 943 015590.
E-mail address: manuel.grana@ehu.es (M. Graña).

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

0925-2312/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2011.02.024

2. Dendritic computing

A single layer morphological neuron endowed with dendrite computation based on lattice algebra was introduced in [21]. Fig. 1 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 (w^0_{ij}, w^1_{ij}) from the synapses coming from the i-th input neuron. The response of the j-th dendrite is as follows:

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

where $l \in L_{ij} \subseteq \{0,1\}$ 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 [21] that models based on dendritic computation have powerful approximation properties. In fact, they 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, which is given in Algorithm 1. Assume that we are given a collection of m pairs of patterns and class labels $(\mathbf{x}^\xi, c_\xi)\mathbf{x}^\xi \in \mathbb{R}^n, c_\xi \in \{0,1\}$. The hard-limiter function of step 3 is the Heaviside function. The algorithm starts building a hyperbox enclosing all pattern samples of class 1, that is, $C_1 = \{\xi : c_\xi = 1\}$. Then, the dendrites are added to the structure trying to remove misclassified patterns of class 0 that fall inside this hyperbox. In step 6 the algorithm selects at random one such misclassified pattern, 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 [2] consists in considering rotations of the patterns obtained from some learning process. Then, the response of the dendrite is given by:

$$\tau_j(\mathbf{x}^\xi) = p_j \bigwedge_{i \in I_j} \bigwedge_{l \in L_{ij}} (-1)^{1-l}(R(\mathbf{x}^\xi)_i + w_{ij}^l), \tag{2}$$

where R denotes the rotation matrix. The process of estimating R can be very time consuming, it is a local process performed during steps 7–10 of the learning process of Algorithm 1. Following this idea, we propose and test in this paper that the enhancement of the single layer neuron model with dendritic computation performance could be obtained from the transformation of the data using the kernel approach in combination with a lattice computing based feature extraction process, the LICA.

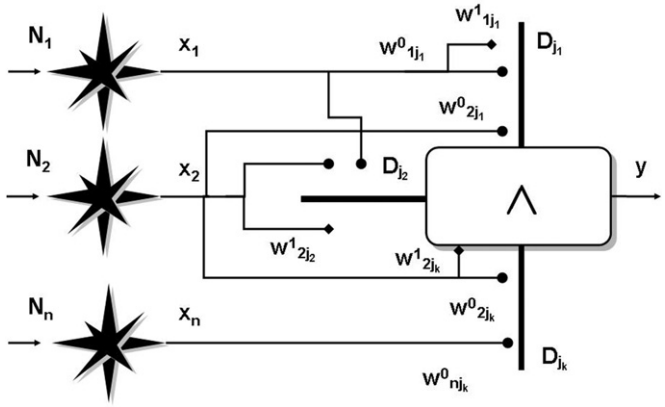

Fig. 1. A single output single layer dendritic computing system.

Algorithm 1. Dendritic computing learning based on elimination.
Training set $T = \{(\mathbf{x}^\xi, c_\xi)\mathbf{x}^\xi \in \mathbb{R}^n, c_\xi \in \{0,1\}; \xi = 1,\ldots,m\}$,

1. Initialize $j=1$, $I_j = \{1,\ldots n\}$, $P_j = \{1,\ldots,m\}$, $L_{ij} = \{0,1\}$,
   $w^1_{ij} = -\bigwedge_{c_\xi = 1} x_i^\xi; w^0_{ij} = -\bigvee_{c_\xi = 1} x_i^\xi, \forall i \in I$
2. Compute response of the current dendrite D_j, with $p_j = (-1)^{\mathrm{sgn}(j-1)}$:
   $\tau_j(\mathbf{x}^\xi) = p_j \bigwedge_{i \in I_j} \bigwedge_{l \in L_{ij}} (-1)^{1-l}(x_i^\xi + w_{ij}^l), \forall \xi \in P_j.$
3. Compute the total response of the neuron:
   $\tau(\mathbf{x}^\xi) = \bigwedge_{k=1}^{j} \tau_k(\mathbf{x}^\xi); \xi = 1,\ldots,m.$
4. If $\forall \xi (f(\tau(\mathbf{x}^\xi)) = c_\xi)$ 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 $f(\tau(\mathbf{x}^\xi)) = 1$.
7. $\mu = \bigwedge_{\xi \neq \gamma} \{ \bigvee_{i=1}^n |x_i^\gamma - x_i^\xi| : \xi \in D \}$.
8. $I' = \{i : |x_i^\gamma - x_i^\xi| = \mu, \xi \in D;\ X = \{(i, x_i^\xi) : |x_i^\gamma - x_i^\xi| = \mu, \xi \in D\}$.
9. $\forall (i, x_i^\xi) \in X$
   (a) if $x_i^\gamma > x_i^\xi$ then $w^1_{ij} = -x_i^\xi$, $E_{ij} = \{1\}$
   (b) if $x_i^\gamma < x_i^\xi$ then $w^0_{ij} = -x_i^\xi$, $H_{ij} = \{0\}$
10. $I_j = I_j \cup I'$; $L_{ij} = E_{ij} \cup H_{ij}$
11. $D' = \{\xi \in D : \forall i \in I_j, -w^1_{ij} < x_i^\xi < -w^0_{ij}\}$. If $D' = \emptyset$ then goto step 2, else $D = D'$ goto step 7.

3. LICA

Lattice Independent Component Analysis is based on the lattice independence discovered when dealing with noise robustness in Morphological Associative Memories [20]. Works on finding lattice independent sources (aka endmembers) for linear unmixing started on hyperspectral image processing [11,16]. Since then, it has been also proposed for functional MRI analysis [10] among other.

Under the Linear Mixing Model (LMM) the design matrix is composed of endmembers which define a convex region covering the measured data. The linear coefficients are known as fractional abundance coefficients that give the contribution of each endmember to the observed data:

$$\mathbf{y} = \sum_{i=1}^{M} a_i \mathbf{s}_i + \mathbf{w} = \mathbf{S}\mathbf{a} + \mathbf{w}, \tag{3}$$

where $\mathbf{y}$ is the d-dimension measured vector, $\mathbf{S}$ is the $d \times M$ matrix whose columns are the d-dimension endmembers $\mathbf{s}_i, i = 1,\ldots,M$, $\mathbf{a}$ is the M-dimension abundance vector, and $\mathbf{w}$ is the d-dimension additive observation noise vector. Under this generative model, two constraints on the abundance coefficients hold. First, to be physically meaningful, all abundance coefficients must be non-negative $a_i \geq 0, i = 1,\ldots,M$, because the negative contribution is not possible in the physical sense. Second, to account for the entire composition, they must be fully additive $\sum_{i=1}^M a_i = 1$. As a side effect, there is a saturation condition $a_i \leq 1, i = 1,\ldots,M$, because no isolate endmember can account for more than the observed material. From a geometrical point of view, these restrictions mean that we expect the endmembers in $\mathbf{S}$ to be an Affine Independent set of points, and that the convex region defined by them covers *all* the data points.

The *Lattice Independent Component Analysis* (LICA) approach assumes the LMM as expressed in Eq. (3). Moreover, the equivalence

between affine independence and strong lattice independence [15] is used to induce from the data the endmembers that compose the matrix **S**. Briefly, LICA consists of two steps:

1. Use an endmember induction algorithm (EIA) to induce from the data a set of strongly lattice independent vectors. In our works we use the algorithm described in [6,7,11,10]. These vectors are taken as a set of affine independent vectors that forms the matrix **S** of Eq. (3).
2. Apply the Full Constrained Least Squares estimation [5,12,13] to obtain the abundance vector according to the conditions for LMM.

The advantages of this approach are (1) that we are not imposing statistical assumptions to find the sources, (2) that the algorithm is one-pass and very fast because it only uses lattice operators and addition, (3) that it is unsupervised and incremental, and (4) that it can be tuned to detect the number of endmembers by adjusting a noise-filtering related parameter. When $M \ll d$ the computation of the abundance coefficients can be interpreted as a dimension reduction transformation, or a feature extraction process. It is under this view that we will use LICA in the experimental works described in Section 5.

3.1. Endmember induction algorithm

The EIA that we introduce in this section is a heuristic that is somehow simpler and faster than the formulations of the EIA algorithm proposed and used in [11,9]. Let us denote $\{\mathbf{x}_i \in \mathbb{R}^d : i=1,\ldots,n\}$ a set of input patterns. Vectors $\vec{\boldsymbol{\mu}}$ and $\vec{\boldsymbol{\sigma}}$ are, respectively, the mean vector and the vector of standard deviations computed component-wise over the data sample, α the filtering factor related to data variability, and E the set of already discovered endmembers. For each input vector, first, the algorithm tests that the input vector is not too similar to the already discovered endmembers, we test for each component independently that the euclidean distance between input and endmembers is lower than the corresponding component $\alpha\vec{\sigma}_j$. The gain parameter α controls the amount of flexibility in the discovering of new endmembers. It determines if a vector is interpreted as a random perturbation of an already selected endmember. It has a great impact on the number of endmembers found, where low values imply large number of endmembers. Lattice independence is tested against the recall provided by the LAAM built from E. strong lattice independence is verified testing the max- or min-dominance on the set of endmembers. The algorithm runs only once over the data. Discussion of its theoretical justification can be found in [11,9] and will not be reproduced here. The detailed description of the steps in the heuristic algorithm is presented as Algorithm 2. In this algorithm we use the symbol $==$ to denote a Matlab-like component-wise vector equality test operator which returns a vector of 0's and 1's, where 0 corresponds to inequality and 1 to equality.

Algorithm 2. A LAAM based incremental endmember induction algorithm.

1. Shift the data sample to zero mean $\{\mathbf{x}_i^c = \mathbf{x}_i - \vec{\boldsymbol{\mu}}; i=1,\ldots,n\}$.
2. Initialize the set of endmembers with the first data sample $E=\{\mathbf{e}^1=\mathbf{x}_1^c\}$. The initial set of endmember sample indices is $I=\{1\}$.
3. Construct the lattice auto-associative memory M_{EE} based on the set of endmembers E.
4. For each input data vector $\mathbf{x}_i^c$
   (a) If there is any $\mathbf{e} \in E$ such that $\forall j : \|\mathbf{x}_{ij}^c - \mathbf{e}_j\| < \alpha\vec{\sigma}_j$ discard $\mathbf{x}_i^c$, otherwise proceed to test SLI
   (b) If $\mathbf{x}_i^c = M_{EE} \boxslash \mathbf{x}_i^c$ then discard $\mathbf{x}_i^c$ because it is lattice dependent on the already discovered endmembers.
   (b) Test max/min-dominance on the enlarged set of endmembers $E' = E \cup \{\mathbf{x}_i^c\}$ to ensure SLI,
      (i) $c_1 = c_2 = 0$
      (ii) for $i=1,\ldots,K+1$
      (iii) $\mathbf{s}_1 = \mathbf{s}_2 = \mathbf{0}$
         A. for $j=1,\ldots,K+1$ and $j \neq i$ $\mathbf{d} = \mathbf{e}_i - \mathbf{e}_j$; $m_1 = \max(\mathbf{d})$; $m_2 = \min(\mathbf{d})$. $\mathbf{s}_1 = \mathbf{s}_1 + (\mathbf{d} == m_1)$, $\mathbf{s}_2 = \mathbf{s}_2 + (\mathbf{d} == m_2)$.
         B. $c_1 = c_1 + (\max(\mathbf{s}_1) == K)$ or $c_2 = c_2 + (\max(\mathbf{s}_2) == K)$.
      (iv) If $c_1 = K+1$ or $c_2 = K+1$ then E' is a set of SLI vectors, go to 3 with the enlarged set of lattice sources and resume exploration with the next input.
5. The output set of endmembers is the set of original data vectors $\{\mathbf{f}(i) : i \in I\}$ corresponding to the vectors selected as members of E.

4. Kernel approaches

The kernel transformation has been found very useful in statistics and pattern recognition applications [24]. A kernel is a function:

$$\kappa(\mathbf{x},\mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle, \tag{4}$$

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:

$$\phi : X \to F. \tag{5}$$

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 [24]) as a PCA of the data in feature space. We have followed the approach as a heuristic, applying also the LICA on the kernel matrix. The obtained success would indicate the need to examine more closely this approach. Finally, we define the Gaussian kernel that will be used in the experiments:

$$\kappa(\mathbf{x},\mathbf{z}) = \exp(-\|\mathbf{x}-\mathbf{z}\|^2/2\sigma^2). \tag{6}$$

5. Experimental results

Fig. 2 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

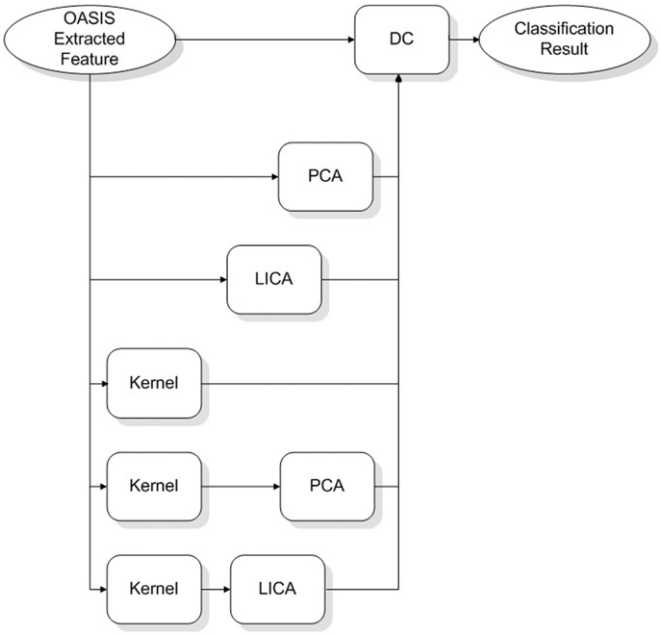

Fig. 2. The experimental exploration.

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 cross-validation creates a random partition into 10 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. 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.

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 Fig. 2 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 $\alpha \in [0.01,0.09] \cup [0.1,0.9] \cup [1,10]$ 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 Fig. 3 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. Fig. 4 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. Fig. 5 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

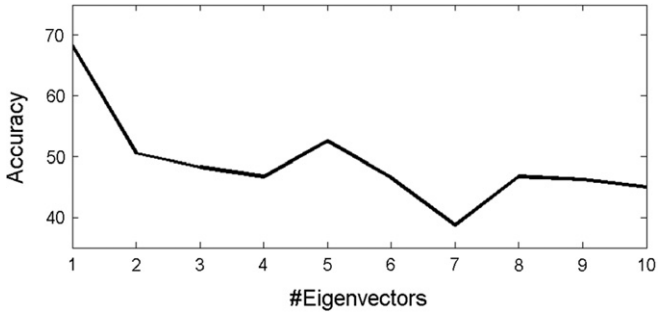

Fig. 3. PCA-DC results as a function of the number of eigenvectors.

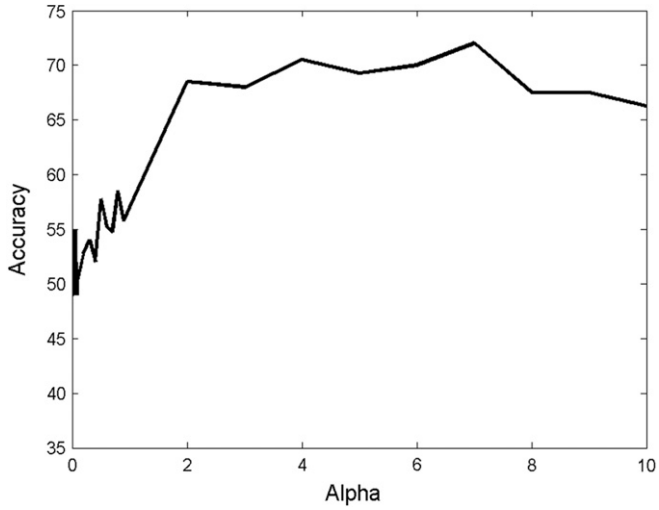

Fig. 4. LICA-DC results as a function of the noise filter parameter α.

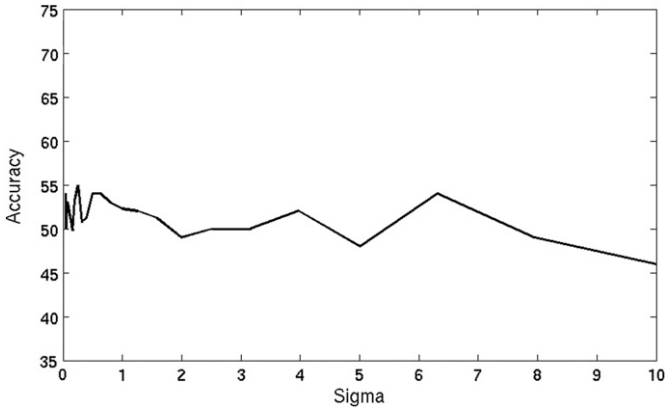

Fig. 5. DC applied to Gaussian Kernel transformation of the data.

kernel and PCA are shown in Fig. 6 as surface depending on the number of eigenvectors selected and the value of the σ parameter. It can be appreciated that 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, Fig. 7 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 Figs. 6 and 7 have embedded the flat surface corresponding to the baseline DC result

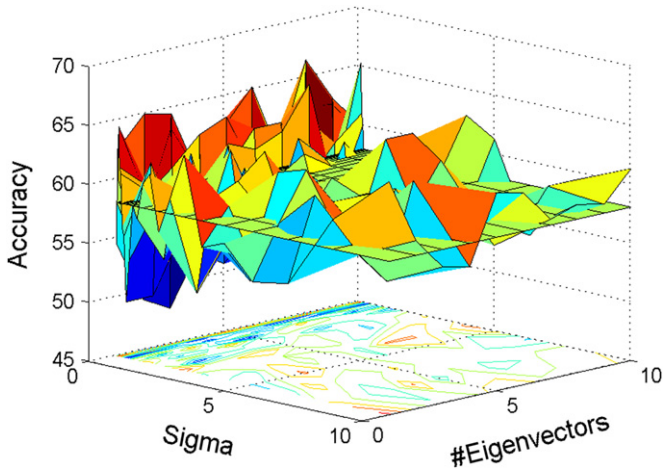

Fig. 6. Kernel-PCA-DC results varying σ and the number of eigenvectors.

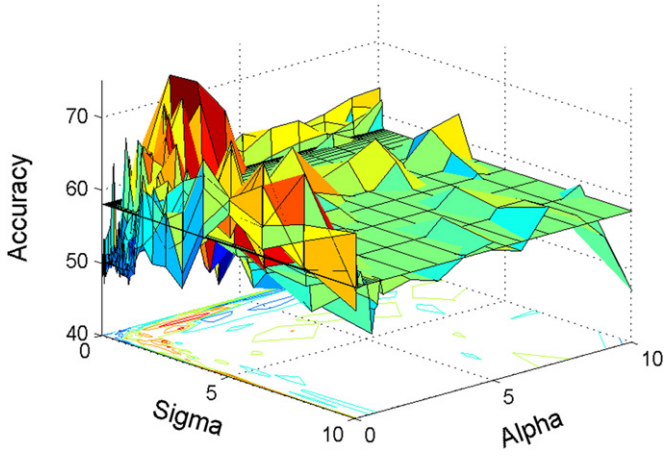

Fig. 7. Kernel-LICA-DC results varying σ and α.

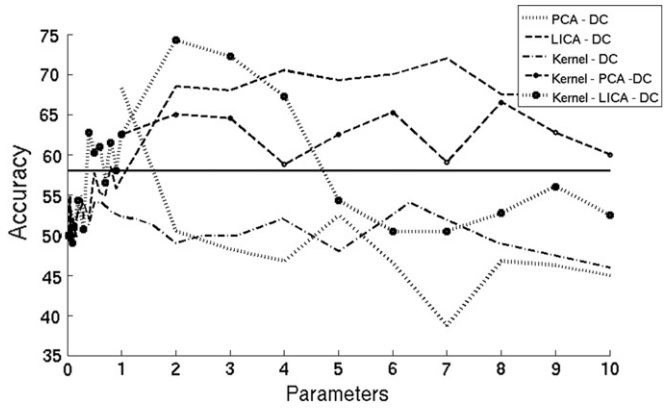

Fig. 8. Comparative plot of the accuracy of all the approaches tested, the meaning of the parameter axis depends on the approach as illustrated in Figs. 3–7.

of 58% accuracy. Therefore the observed peaks correspond to parameter combinations where the combination of systems improves the baseline DC.

Fig. 8 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. 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

Table 1
Summary of best results of validation experiments over AD feature database.

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

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 1 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 [4,22], 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.

6. Conclusions

We found empirically, performing cross-validation on an Alzheimer's Disease database of features extracted from MRI scans that a single layer neuron model endowed with dendritic computing has poor generalization capabilities. The model shows high sensitivity but poor specificity. In this paper we have proposed the application of a composition of processes to enhance the model generalization properties. Specifically, we propose to perform a 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.

References

[1] A. Abraham, E. Corchado, J.M. Corchado, Hybrid learning machines, Neurocomputing 72 (13–15) (2009) 2729–2730.
[2] A. Barmpoutis, G.X. Ritter, Orthonormal basis lattice neural networks, in: IEEE International Conference on Fuzzy Systems, 2006, 2006, pp. 331–336.
[3] E. Corchado, A. Abraham, A.C.P.L. Ferreira de Carvalho, Hybrid intelligent algorithms and applications, Information Sciences 180 (14) (2010) 2633–2634.
[4] M. García-Sebastián, A. Savio, M. Graña, J. Villanúa, On the use of morphometry based features for Alzheimer's disease detection on MRI, in: J. Cabestany, F. Sandoval, A. Prieto, J.M. Corchado (Eds.), Bio-Inspired Systems: Computational and Ambient Intelligence/IWANN 2009 (Part I), Lecture Notes in Computer Science, vol. 5517, 2009, pp. 957–964.
[5] G.H. Golub, C.F. VanLoan, Matrix Computations, The Johns Hopkins University Press, 1996.
[6] M. Graña, J. Gallego, Associative Morphological Memories for endmember induction, Proceedings of 2003 IEEE International Geoscience and Remote Sensing Symposium, 2003. IGARSS '03, vol. 6, 2003, pp. 3757–3759.
[7] M. Graña, P. Sussner, G.X. Ritter, Associative morphological memories for endmember determination in spectral unmixing, The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03, vol. 2, 2003, pp. 1285–1290 May.
[8] M. Graña, A brief review of lattice computing, in: Proceedings of the WCCI 2008, 2008, pp. 1777–1781.
[9] M. Graña, D. Chyzhyk, M. García-Sebastián, C. Hernández, Lattice independent component analysis for functional magnetic resonance imaging, Information Sciences 181 (2011) 1910–1928.
[10] M. Graña, A. Manhaes-Savio, M. García-Sebastián, E. Fernandez, A lattice computing approach for on-line fMRI analysis, Image and Vision Computing 28 (7) (2010) 1155–1161.

[11] M. Graña, I. Villaverde, J.O. Maldonado, C. Hernandez, Two lattice computing approaches for the unsupervised segmentation of hyperspectral images, Neurocomputing 72 (10–12) (2009) 2111–2120.
[12] D. Heinz, C.-I. Chang, Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery, IEEE Transactions on Geoscience and Remote Sensing 39 (2000) 529–545.
[13] C.L. Lawson, H.J. Hanson, Solving Least Squares Problems, Prentice-Hall, Englewoods Cliffs, NJ, 1974.
[14] Gerhard Ritter, Paul Gader, Fixed points of lattice transforms and lattice associative memories, Advances in Imaging and Electron Physics, vol. 144, Elsevier, 2006, pp. 165–242.
[15] G.X. Ritter, G. Urcid, A lattice matrix method for hyperspectral image unmixing, Information Sciences 181 (2011) 1787–1803.
[16] G.X. Ritter, G. Urcid, M.S. Schmalz, Autonomous single-pass endmember approximation using lattice auto-associative memories, Neurocomputing 72 (10–12) (2009) 2101–2110 (Lattice Computing and Natural Computing (JCIS 2007)/Neural Networks in Intelligent Systems Design (ISDA 2007)).
[17] G.X. Ritter, L. Iancu, Single layer feedforward neural network based on lattice algebra, Proceedings of the International Joint Conference on Neural Networks, 2003, vol. 4, 2003, pp. 2887–2892 July.
[18] G.X. Ritter, L. Iancu, A morphological auto-associative memory based on dendritic computing, Proceedings of the IEEE International Joint Conference on Neural Networks 2004, vol. 2, 2004, pp. 915–920 July.
[19] G.X. Ritter, L. Iancu, G. Urcid, Morphological perceptrons with dendritic structure, The 12th IEEE International Conference on Fuzzy Systems, 2003. FUZZ '03, vol. 2, 2003, pp. 1296–1301 May.
[20] G.X. Ritter, P. Sussner, J.L. Diaz de Leon, Morphological associative memories, IEEE Transactions on Neural Networks 9 (2) (1998) 281–293.
[21] G.X. Ritter, G. Urcid, Lattice algebra approach to single-neuron computation, IEEE Transactions on Neural Networks 14 (2) (2003) 282–295.
[22] A. Savio, M. García-Sebastián, M. Graña, J. Villanúa, Results of an adaboost approach on Alzheimer's disease detection on MRI, in: J. Mira, J.M. Ferrández, J.R. Alvarez, F. dela Paz, F.J. Tolede (Eds.), Bioinspired Applications in Artificial and Natural Computation, Lecture Notes in Computer Science, vol. 5602, 2009, pp. 114–123.
[23] A. Savio, M. García-Sebastián, C. Hernández, M. Graña, J. Villanúa, Classification results of artificial neural networks for Alzheimer's Disease detection, in: Emilio Corchado, Hujun Yin (Eds.), Intelligent Data Engineering and Automated Learning-IDEAL 2009, Lecture Notes in Computer Science, vol. 5788, 2009, pp. 641–648.
[24] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, 2004.

Darya Chyzyk is a Ph.D. Student at the Computer Science Department of the Universidad del Pais Vasco. Her research interest include medical image processing, artificial neural networks architectures and applications. She has co-edited one book, published one journal paper and several conference papers.

Manuel Graña is full professor at the Computer Science Department of the Universidad del Pais Vasco. His research interest include image processing, artificial neural networks architectures and applications, robotics and computer vision. He has co-edited several books, published more than 50 journal papers and more than a hundred conference papers.

Alexandre Savio is a Ph.D. Student at the Computer Science Department of the Universidad del Pais Vasco. His research interest include medical image processing, artificial neural networks architectures and applications. He has co-edited one book, published two journal papers and more than 10 conference papers.

Josu Maiora is an associate professor and Ph.D. student at the Computer Science Department of the Universidad del Pais Vasco. His research interest include medical image processing, artificial neural networks architectures and applications.