Optimal Hyperbox shrinking in Dendritic Computing applied to Alzheimer's Disease detection in MRI

Darya Chyzhyk and Manuel Graña

1 Introduction

Dendritic Computing (DC) [1, 3, 5, 6, 7] 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 [4, 7]. 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 [2, 8, 9, 10]. 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 3.

In previous computational experiments we have noticed that the SNLDC results in high sensitivity and very low specificify. We atribute this to the fact that the learning algorithm always tries to guarantee the good classification of the class 1 samples. In this paper we propose to apply a reduction factor on the size of the hyperboxes created by the SNLDC learning algorithm. The results show a better trade off between sensitivity and specificity, increasing the classifier accuracy.

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 [9, 8, 2]. We selected a balanced set of AD patients and controls of the same sex, 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 we use the

Grupo de Inteligencia Computacional, UPV/EHU, www.ehu.es/ccwintco

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

mean and standard deviation of the voxel values of these clusters. FIgure 1 shows the pipeline of the processes performed up to the classification with the DC system.

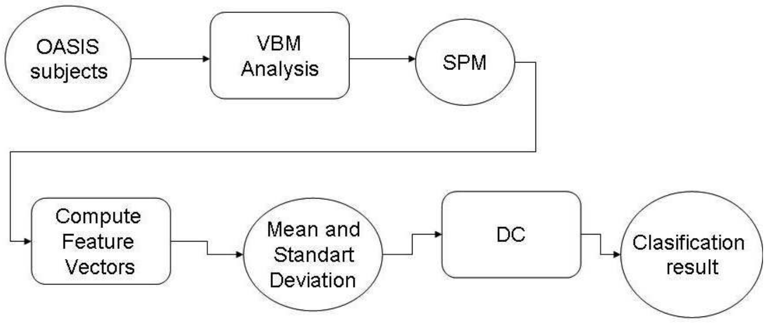

Fig. 1 Pipeline of the process performed, including VBM, feature extraction and classification by DC

The structure of the paper is the following. Section 2 introduces the DC classification system and the training algorithm. Section 3 gives our experimental results on the AD database. Section 4 gives our conclusions.

2 Dendritic Computing

A single layer morphological neuron endowed with dendrite computation based on lattice algebra was introduced in [7]. Figure 2 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 $\left(w_{ij}^0, w_{ij}^1\right)$ from the synapses coming from the *i*-th input neuron. The response of the j-th dendrite is as follows:

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

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 [7] 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. The hard-limiter function of step 3 is the signum function. The algorithm starts building a hyperbox enclosing all pattern samples of

Algorithm 1 Dendritic Computing learning based on elimination

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

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

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

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

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

3. Compute the total response of the neuron:

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

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 $j = j+1$, $I_j = I' = X = E = H = \varnothing$, $D = C_1$
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 $D' = \varnothing$ then goto step 2, else $D = D'$ goto step 7.

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 [1] 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 1.

In this paper we will 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 1. The effect of this strategy can be appreciated comparing figures 3 and 4. In figure 3 we show the boxes generated by the original learning algorithm. Objects of class 1 correspond to crosses. In figure 4 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.

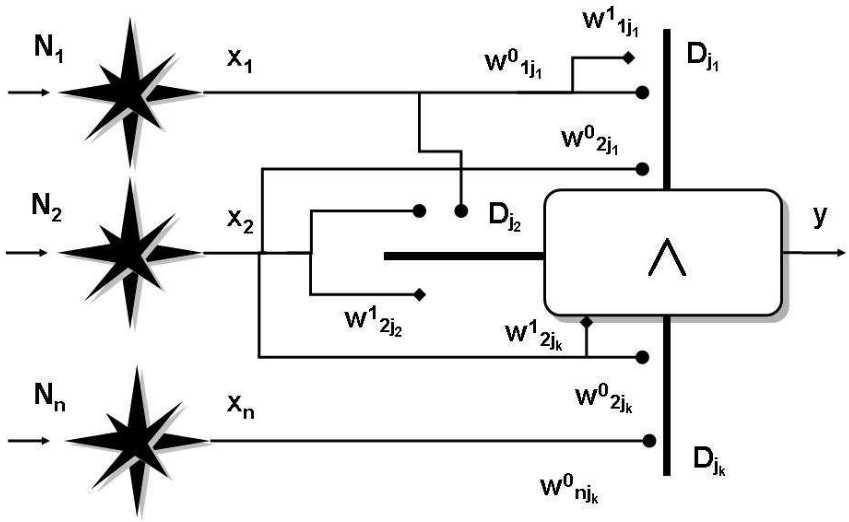

Fig. 2 A single ouput single layer Dendritic Computing system.

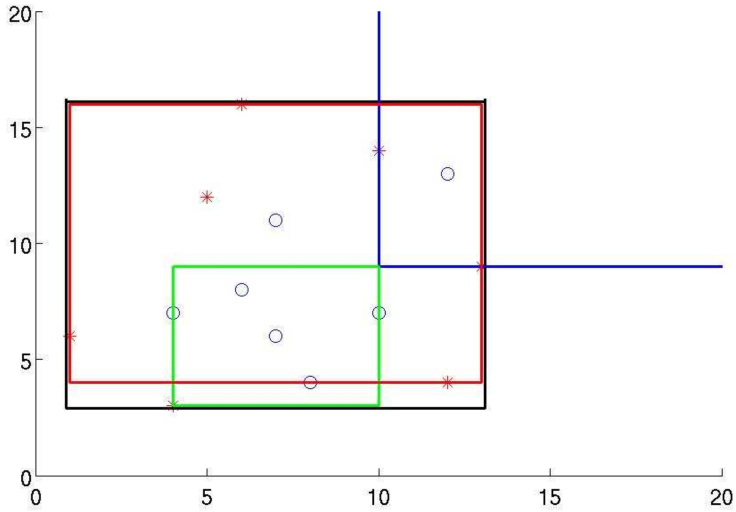

Fig. 3 Resulting boxes of the original DC learning on a synthetic 2D dataset

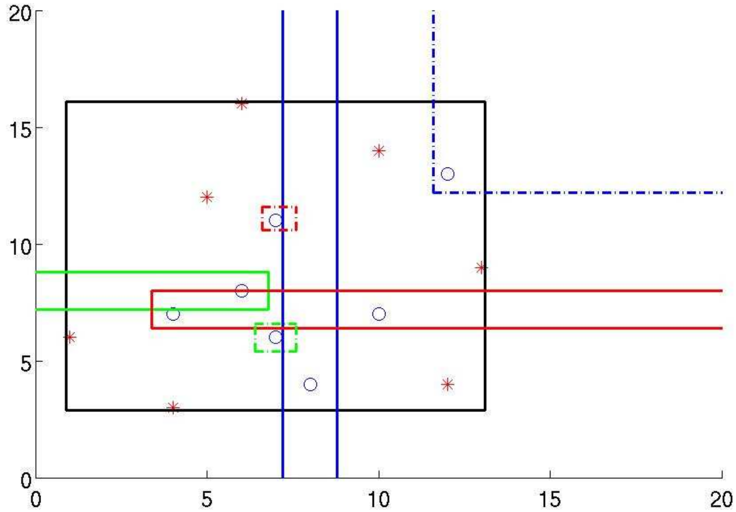

Fig. 4 Resulting boxes of the DC algorithm with shrinking factor $\alpha = 0.8$.

3 Experimental results

For each shinking 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 1 and Figure 5 where the first row corresponds to the baseline DC algorithm. 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 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[2, 10], while the specificity is still below the results obtained by other state of art approaches.

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 1 Summary of best results of validation experiments over AD MSD feature database. First row corresponds to the original DC algorithm[7].

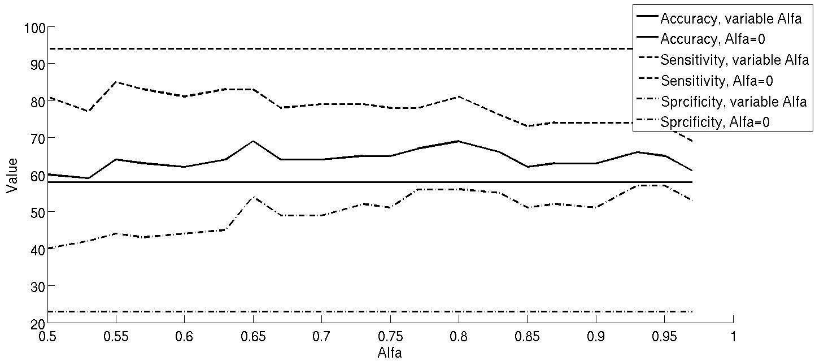

Fig. 5 DC result varing α and $\alpha = 0$

4 Conclusions

We found empirically, performing cross-validation on an Alzheimer's Disease database of features computed 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 a simple change in the learning algorithm that produces a significative increase in performance in terms of accuracy, obtaining a better trade-off between sensitivity and specificity. This strategy could be combined with other techniques to enhance further the performance of DC.

References

1. A. Barmpoutis and G.X. Ritter. Orthonormal basis lattice neural networks. In *Fuzzy Systems, 2006 IEEE International Conference on*, pages 331 –336, 2006.
2. M. García-Sebastián, A. Savio, M. Graña, and J. Villanúa. On the use of morphometry based features for alzheimer's disease detection on mri. *Bio-Inspired Systems: Computational and Ambient Intelligence. / IWANN 2009 (Part I) Joan Cabestany, Francisco Sandoval, Alberto Prieto, Juan M. Corchado (Editors), LNCS 5517*, pages 957–964, 2009.
3. Gerhard Ritter and Paul Gader. Fixed points of lattice transforms and lattice associative memories. volume 144 of *Advances in Imaging and Electron Physics*, pages 165 – 242. Elsevier, 2006.
4. G.X. Ritter and L. Iancu. Single layer feedforward neural network based on lattice algebra. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 4, pages 2887 – 2892 vol.4, jul. 2003.
5. G.X. Ritter and L. Iancu. A morphological auto-associative memory based on dendritic computing. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 915 – 920 vol.2, jul. 2004.
6. G.X. Ritter, L. Iancu, and G. Urcid. Morphological perceptrons with dendritic structure. In *Fuzzy Systems, 2003. FUZZ '03. The 12th IEEE International Conference on*, volume 2, pages 1296 – 1301 vol.2, may. 2003.

7. G.X. Ritter and G. Urcid. Lattice algebra approach to single-neuron computation. *Neural Networks, IEEE Transactions on*, 14(2):282 – 295, mar. 2003.
8. A. Savio, M. García-Sebastián, M. Graña, and J. Villanúa. Results of an adaboost approach on alzheimer's disease detection on mri. *Bioinspired applications in Artificial and Natural Computation. J. Mira, J. M. Ferrández, J.R. Alvarez, F. dela Paz, F.J. Tolede (Eds.) LNCS 5602*, pages 114–123, 2009.
9. A. Savio, M. García-Sebastián, C. Hernández, M. Graña, and J. Villanúa. Classification results of artificial neural networks for alzheimer's disease detection. *Intelligent Data Engineering and Automated Learning- IDEAL 2009, Emilio Corchado, Hujun Yin (eds) LNCS 5788*, pages 641–648, 2009.
10. A. Savio, M.T Garcia-Sebastian, D. Chyzhyk, C. Hernandez, M. Grana, A. Sistiaga, A. Lopez de Munain, and J. Villanua. Neurocognitive disorder detection based on feature vectors extracted from vbm analysis of structural mri. *Computers in Biology and Medicine*, accepted with revisions, 2011.