eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea

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

Darya Chyzhyk, Manuel Graña

Computational Intelligence Group
Dept. CCIA,
University of Basque Country,
San Sebastián, Spain
www.ehu.es/ccwintco

Content

- Introduction
- Dendritic Computing
- Experimental results
- Summary and Conclusions

Introduction

Dendritic Computing

- simple and fast
- based on biology
- binary class problems
- based on lattice theory

Introduction

Has been proved to compute a **perfect** approximation to **any data distribution.**

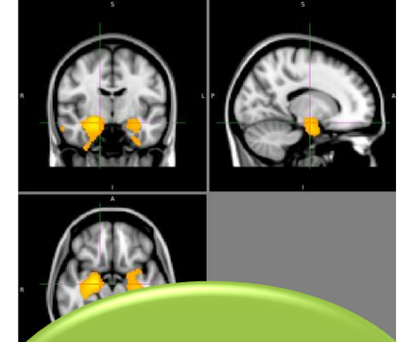
The results of cross-validation experiments give very poor performance: **high sensitivity and very low specificity**.

We attribute this to the fact that the **DC learning algorithm** always tries to **guarantee** the good classification of the class 1 samples.

We propose to apply a **reduction factor** on the size of the hyperboxes

OASIS subjects →

VBM Analysis →

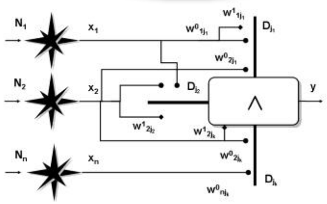


SPM

↓

Compute Feature Vectors ←

Mean and standard Deviation ←



DC

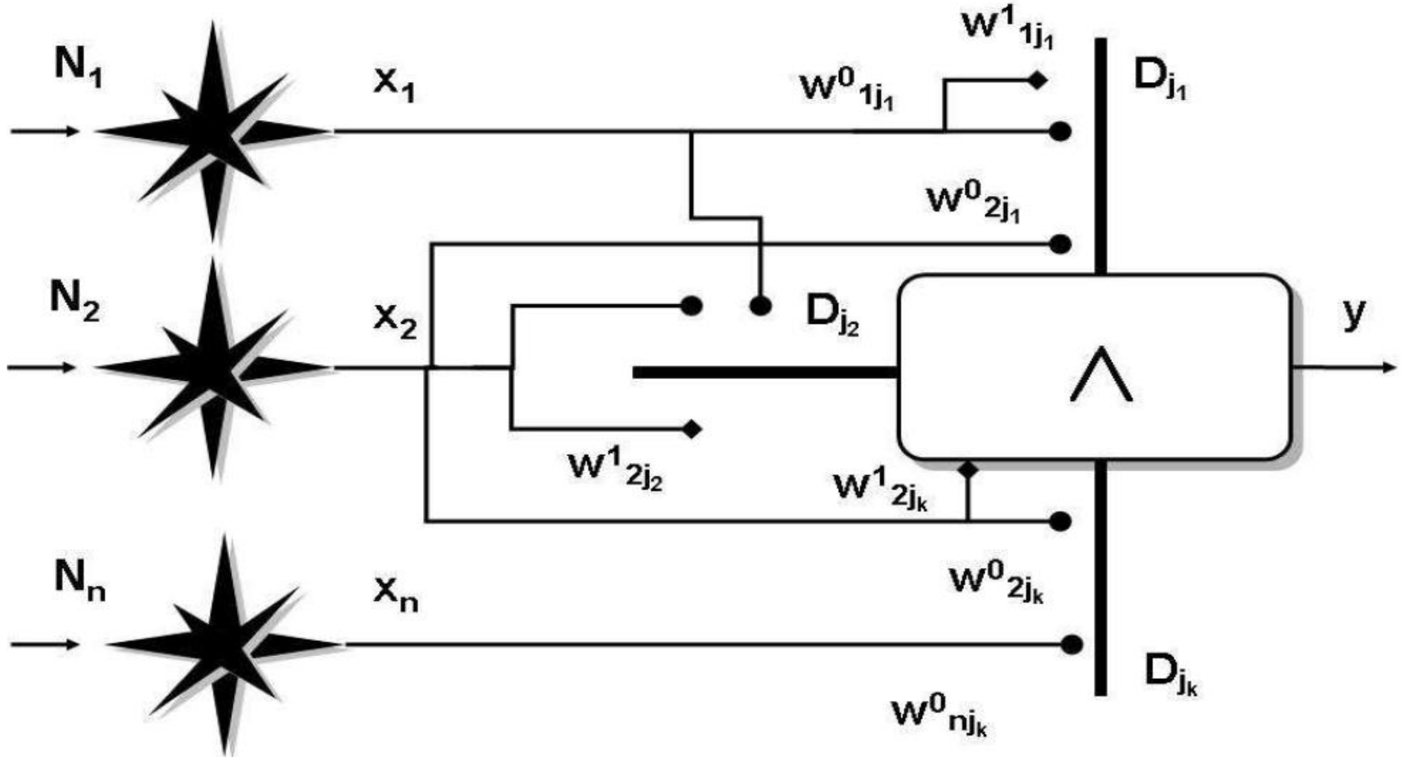
↓

Result

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

Dendritic Computing

Structure of a single output class single layer Dendritic Computing system



Algorithm 1 Dendritic Computing learning based on elimination

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

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

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

Construct a basic hyperbox

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:

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

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 $f(\tau(\mathbf{x}^{\gamma})) = 1$.
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.

Algorithm 1 Dendritic Computing learning based on elimination

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

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

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

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:

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

Hard-limiter function

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 $f(\tau(\mathbf{x}^{\gamma})) = 1$.
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.

Algorithm 1 Dendritic Computing learning based on elimination

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

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

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

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 $f(\tau(\mathbf{x}^{\gamma})) = 1$.
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. $\forall \left(i, x_i^{\xi} \right) \in X$

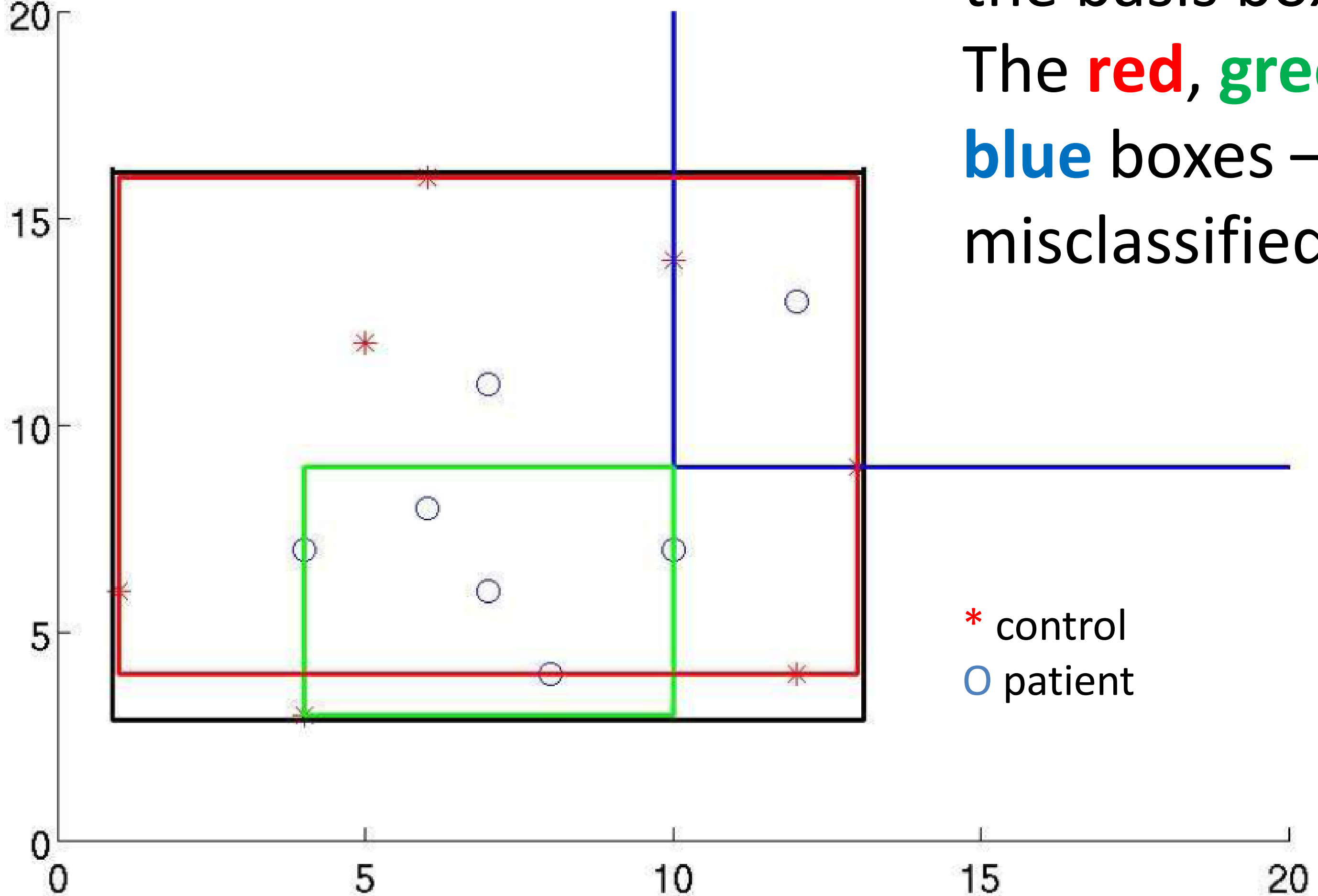
   a. if $x_i^{\gamma} > x_i^{\xi}$ then $w_{ij}^{1} = -(x_i^{\xi} + \alpha \cdot \mu)$, $E_{ij} = \{1\}$
   b. if $x_i^{\gamma} < x_i^{\xi}$ then $w_{ij}^{0} = -(x_i^{\xi} - \alpha \cdot \mu)$, $H_{ij} = \{0\}$

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.

Adding of dendrites to remove misclassified patterns of class 0 that fall inside this hyperbox

Dendritic Computing

Synthetic 2D dataset.

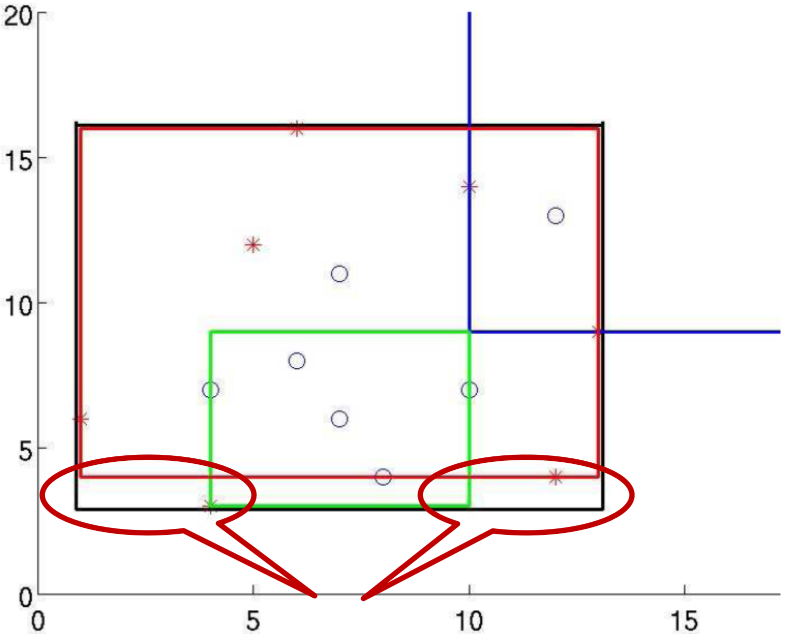


The **black** hyperbox is the basis box.
The **red**, **green** and **blue** boxes – to remove misclassified controls

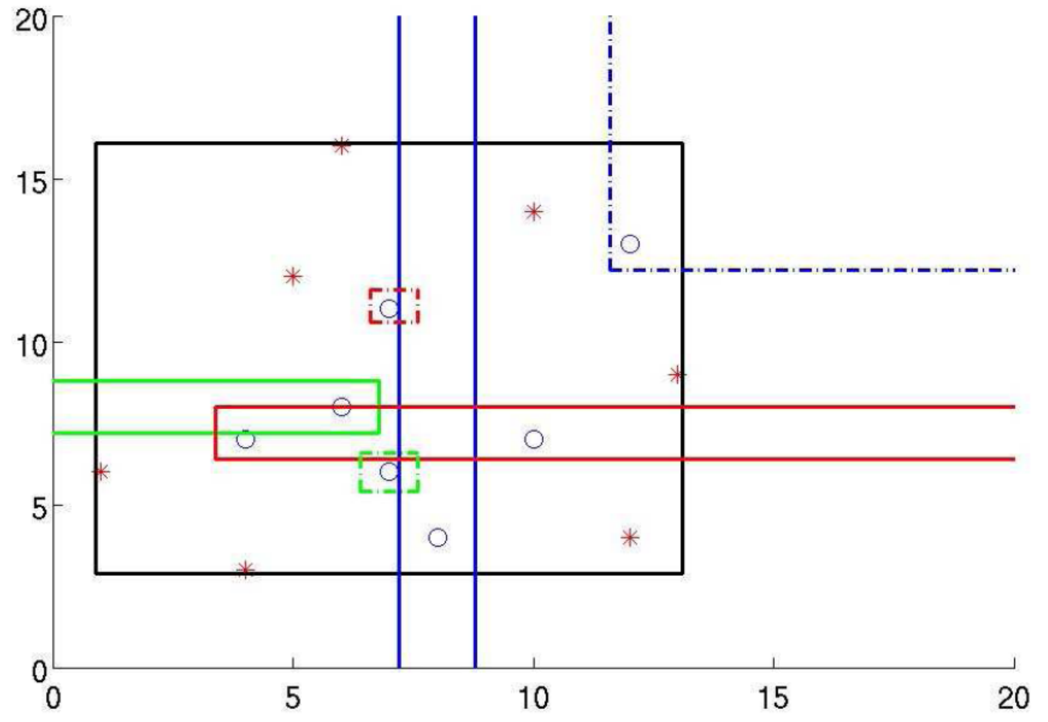
* control
O patient

Dendritic Computing

- To establish specificity and sensitivity we propose shrinking the boundaries of the hyperbox.
- Exclude the region occupied by a misclassified item of control

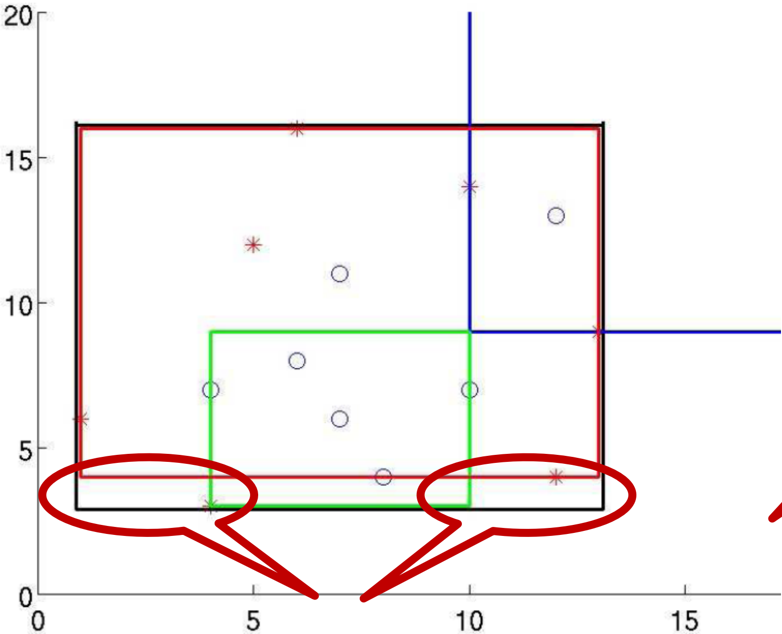


Generalization for patient

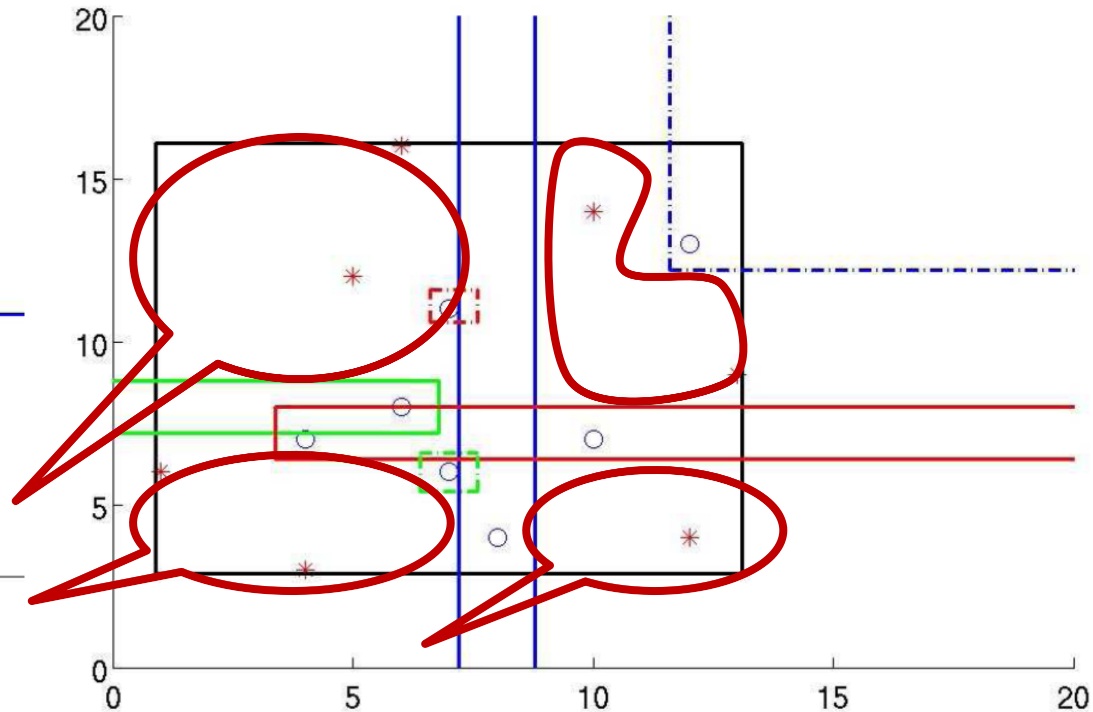


Dendritic Computing

- To balance specificity and sensitivity we propose shrinking the boundaries of the hyperbox corresponding to each dendrite
- Exclude the region occupied by a misclassified item of control



Generalization for patient



Results on Alzheimer Detection

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

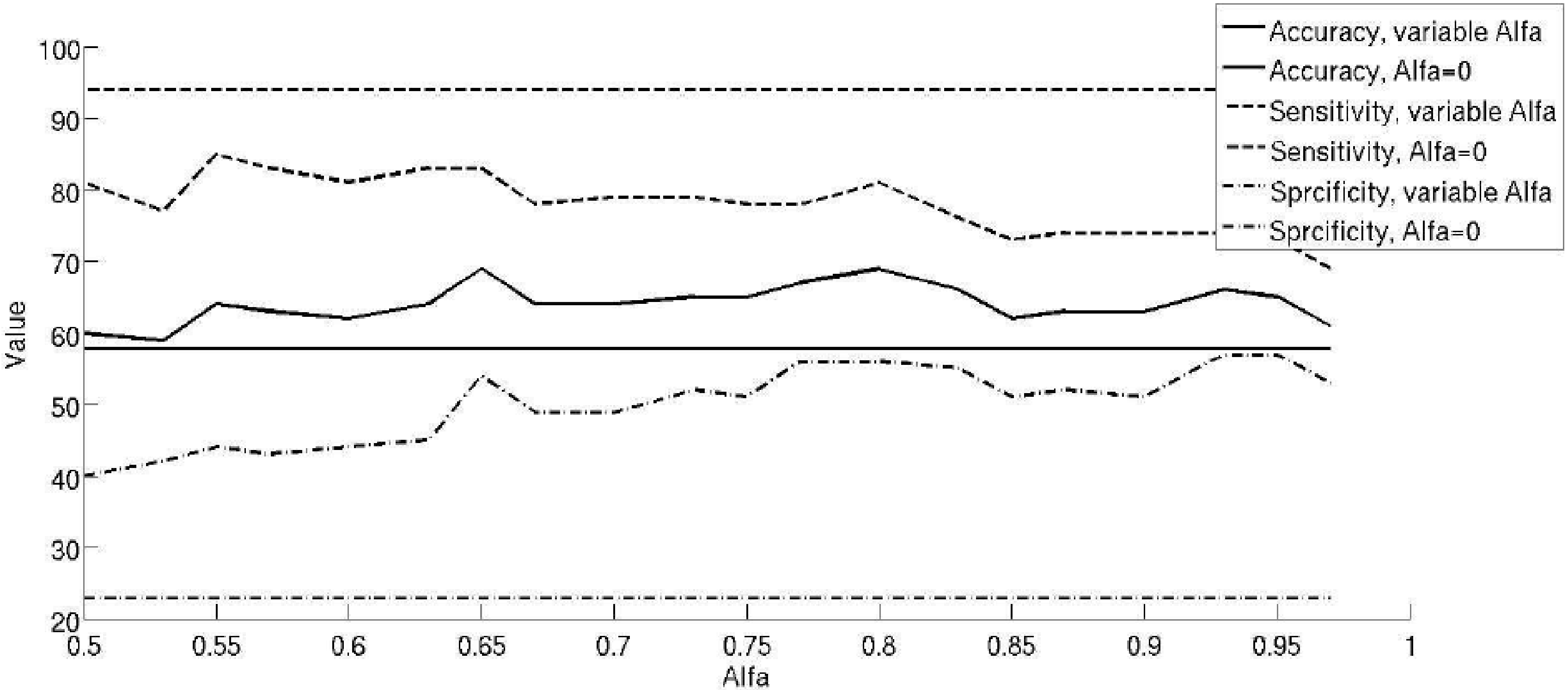
First row – **baseline DC**

We defind Shrinking parameter of the box *αϵ[0,1).*

For each shrinking parameter we have performed 10 –fold cross validation

The best result:
Sensitivity worse
Specificity **increase** – the best balance which gives the best Accuracy

Results on Alzheimer Detection



- **Specificity** (at the bottom) – control classification:

For 0.5 increase very fast and still increase

- **Sensitivity** (at the top) – patient classification:

Shows the better results for baseline DC

- But the **Accuracy** (in the middle)

Shows the best result for *α=0.8*

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.

Thank you for your attention!

Darya Chyzhyk, Manuel Graña

Computational Intelligence Group
University of Basque Country,
San Sebastián, Spain
www.ehu.es/ccwintco