New information fusion techniques and applications on image processing, classification and the computational brain

Javier Fernandez

Universidad Pública de Navarra

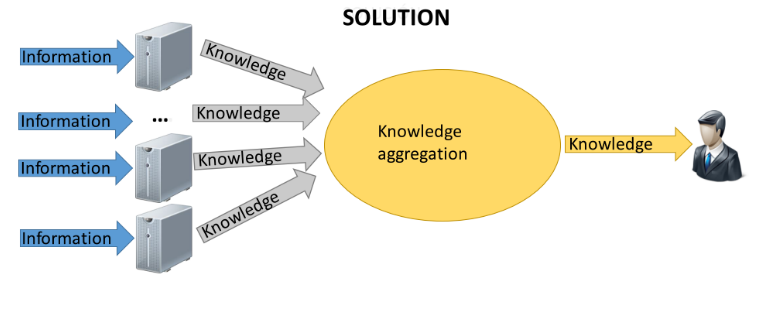
Kempten 2020
Nov 20, 2020

Index

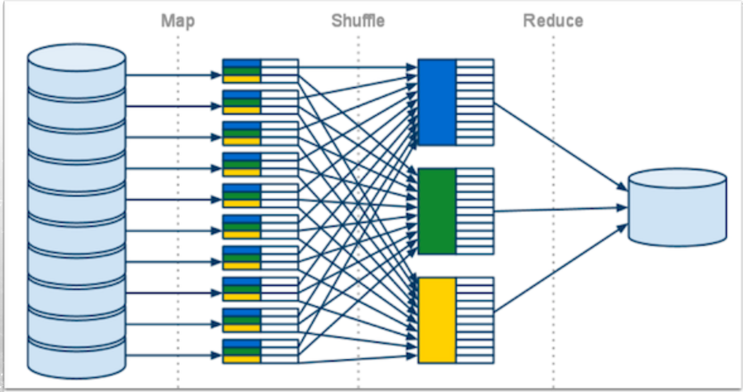
Index

1. The necessity of fusing data
2. Data fusion functions
3. Pre-aggregations
4. The computational brain
5. Conclusions

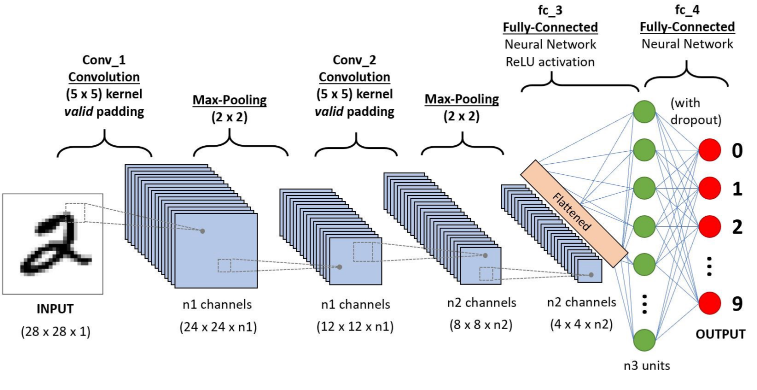
Too much information



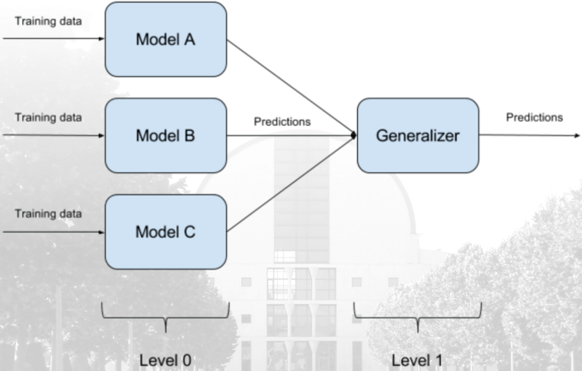
Hadoop for Big Data



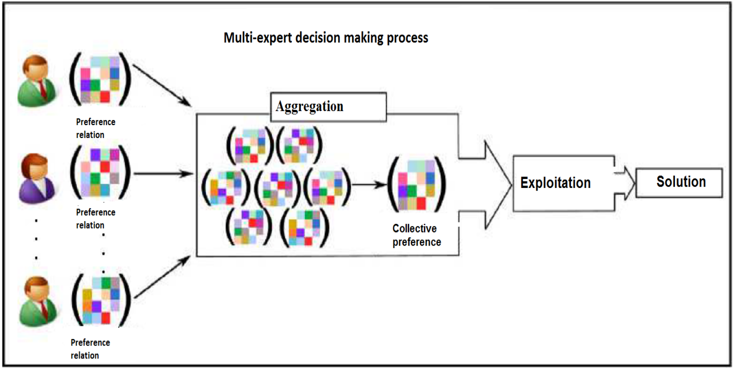
Deep Learning



Problem: How do we choose the architecture
Solution: Use different architectures and fuse the results (ensembles)



Decision making



Ensembles

An inspiring sentence

> Science is made of data, as a house is made of bricks. But a set of data is not science, in the same way as a set of bricks is not a house.

H. Poincaré

Index

1. Introduction
2. Data fusion functions
3. Pre-aggregations
4. The computational brain
5. Conclusions

A simple idea

Definition

Let $n \geq 2$. We call fusion function to any function $F : [0,1]^n \to [0,1]$.

We represent a set of data by one single value of the same nature.

- Unit interval $\Rightarrow$ Not relevant.

A simple idea

Definition

Let $n \geq 2$. We call fusion function to any function $F : [0,1]^n \to [0,1]$.

We represent a set of data by one single value of the same nature.

- Unit interval $\Rightarrow$ Not relevant.
- Other conditions $\Rightarrow$ None.

A VERY GENERAL definition

Imposing conditions: monotonicity

Definition

A function $F : [a,b]^n \to [a,b]$ *is increasing if for each* $x_1, \ldots, x_n$, $y_1, \ldots, y_n \in [a,b]$ *with* $x_i \leq y_i$ *for every* $i = 1, \ldots, n$*, it holds that:*

$$F(x_1, \ldots, x_n) \leq F(y_1, \ldots, y_n).$$

Imposing conditions: Aggregation functions

Definition

An aggregation function is a fusion function $M : [0,1]^n \to [0,1]$ *such that:*

1. M *is increasing;*
2. $M(0,\ldots,0) = 0$*;*
3. $M(1,\ldots,1) = 1$.

Imposing conditions: Aggregation functions

Definition

An aggregation function is a fusion function $M : [0,1]^n \to [0,1]$ *such that:*

1. M *is increasing;*
2. $M(0, \ldots, 0) = 0$*;*
3. $M(1, \ldots, 1) = 1$*.*

Definition

An aggregation function M *is idempotent if, for every* $t \in [0,1]$*,* $M(t, \cdots, t) = t$*.*

Relevant examples

- Our definition does not take into account, in principle, relations between data.

- We want to take such relation explicitely into account.

Linking data: Fuzzy measures

Definition

Let $N = \{1, \cdots, n\}$*. A function* $\mathfrak{m} : 2^N \to [0,1]$ *is a discrete fuzzy measure if, for all* $X, Y \subseteq N$*, it satisfies the following properties:*

($\mathfrak{m}1$) *Increasingness: if* $X \subseteq Y$*, then* $\mathfrak{m}(X) \leq \mathfrak{m}(Y)$*;*

($\mathfrak{m}2$) *Boundary conditions:* $\mathfrak{m}(\emptyset) = 0$ *and* $\mathfrak{m}(N) = 1$*.*

The key example: Choquet integral

Definition

Let $\mathfrak{m} : 2^N \to [0,1]$ be a fuzzy measure. The discrete Choquet integral of $\mathbf{x} = (x_1, \ldots, x_n) \in [0,1]^n$ with respect to $\mathfrak{m}$ is defined as a function $C_{\mathfrak{m}} : [0,1]^n \to [0,1]$, given by

$$C_{\mathfrak{m}}(\mathbf{x}) = \sum_{i=1}^{n} \left(x_{(i)} - x_{(i-1)}\right) \cdot \mathfrak{m}\left(A_{(i)}\right), \tag{1}$$

where $\left(x_{(1)}, \ldots, x_{(n)}\right)$ is an increasing permutation on the input $\mathbf{x}$, that is, $0 \leq x_{(1)} \leq \ldots \leq x_{(n)}$, with the convention that $x_{(0)} = 0$, and $A_{(i)} = \{(i), \ldots, (n)\}$ is the subset of indices of the $n - i + 1$ largest components of $\mathbf{x}$.

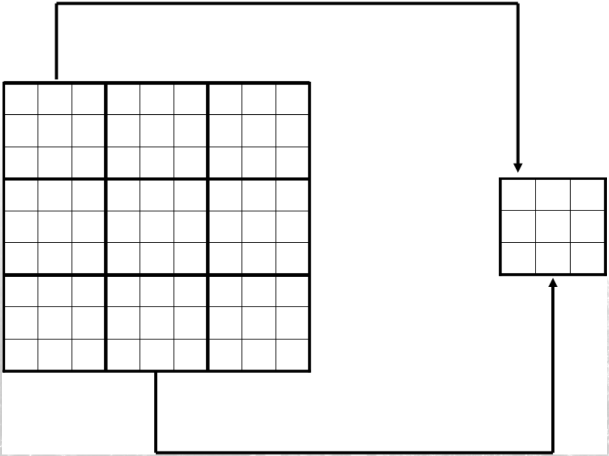
The Choquet integral is a continuous piecewise linear idempotent aggregation function

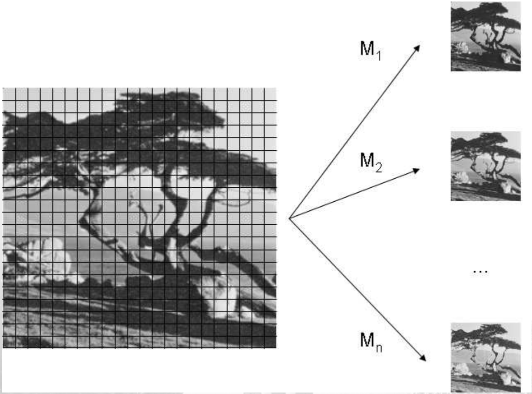
An interesting question

How can we determine the best aggregation function for a given problem??

Trial or...

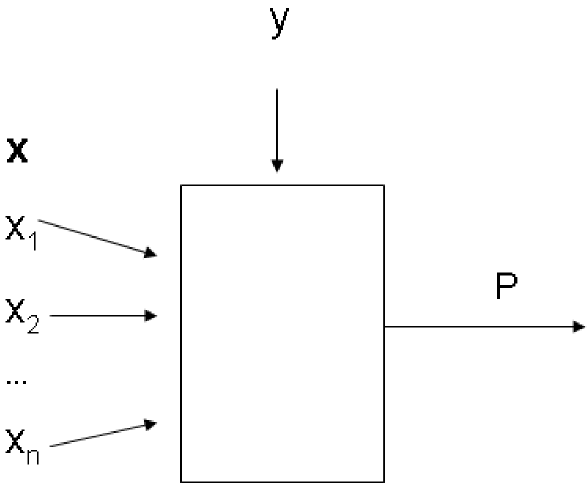
Image processing. Reduction





Construction of image reduction operators using averaging aggregation functions. D. Paternain, J. Fernandez, H. Bustince, R. Mesiar, G. Beliakov Fuzzy Sets and Systems, 261, 87-111 (2015)

What have we done



Definition

A penalty function is a mapping

$$P : [a, b]^{n+1} \to \mathbb{R}^+ = [0, \infty]$$

such that:

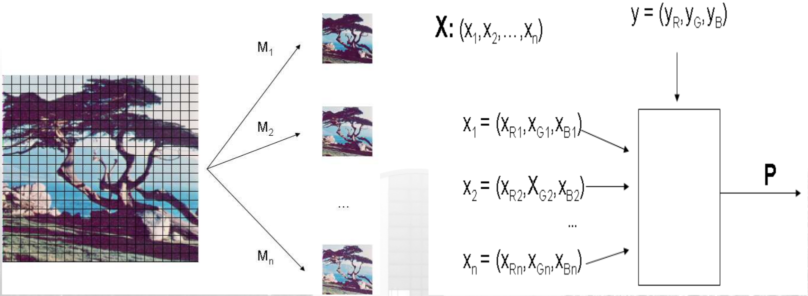
1. $P(\mathbf{x}, y) = 0$ *if* $x_i = y$ *for every* $i = 1, \cdots, n$*;*
2. $P(\mathbf{x}, y)$ *is quasi-convex in* y *for every* $\mathbf{x}$*; that is,*

$$P(\mathbf{x}, \lambda \cdot y_1 + (1 - \lambda) \cdot y_2) \leq \max(P(\mathbf{x}, y_1), P(\mathbf{x}, y_2))$$

Aggregation functions based on penalties. Tomasa Calvo, Gleb Beliakov, Fuzzy Sets and Systems, 161 (10), 1420-1436 (2010)

On the definition of penalty functions in data aggregation. Humberto Bustince, Gleb Beliakov, Gracaliz Pereira Dimuro, Benjamin Bedregal, Radko Mesiar, Fuzzy Sets and Systems, 323 (15), 1-18 (2017)

Penalty functions and further



Penalty functions in a Cartesian product of lattices

Definition

A function $P_\nabla : ([0,1]^n)^m \times [0,1]^m \to [0,\infty[$ is a penalty function if, for every $\mathbf{X} = (\mathbf{x^1}, \ldots, \mathbf{x^m}) \in ([0,1]^n)^m$ (with $\mathbf{x^i} = (x_1^i, \ldots, x_n^i)$ for every $i \in \{1, \ldots, m\}$) and for every $\mathbf{y} = (y_1, \ldots, y_m) \in [0,1]^m$, it satisfies that:

1. $P_\nabla(\mathbf{X}, \mathbf{y}) \geq 0$;
2. $P_\nabla(\mathbf{X}, \mathbf{y}) = 0$ if and only if $x_1^i = \cdots = x_n^i = y^i$ for every $i \in \{1, \ldots, m\}$;
3. P_∇ is convex in y_i or every $i \in \{1, \ldots, m\}$.

$$P_\nabla(\mathbf{X}, \mathbf{y}) = \sum_{q=1}^{m} \sum_{p=1}^{n} |x_p^q - y_q|^2$$

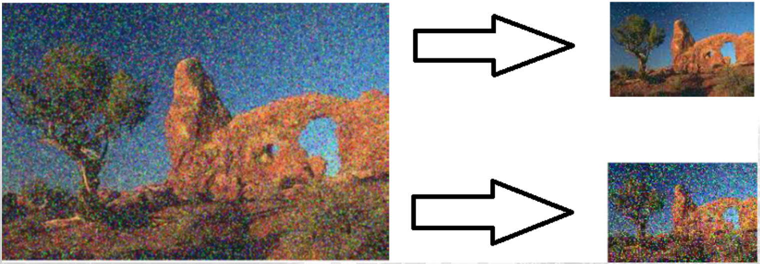
Consensus in multi-expert decision making problems using penalty functions defined over a Cartesian product of lattices. H. Bustince, E. Barrenechea, T. Calvo, S. James, G. Beliakov, Information Fusion, 17, 56-64 (2014)

Set of aggregation functions ($q = 5$ chosen 3 by 3): $\wedge, \vee$, geometric mean, arithmetic mean, median

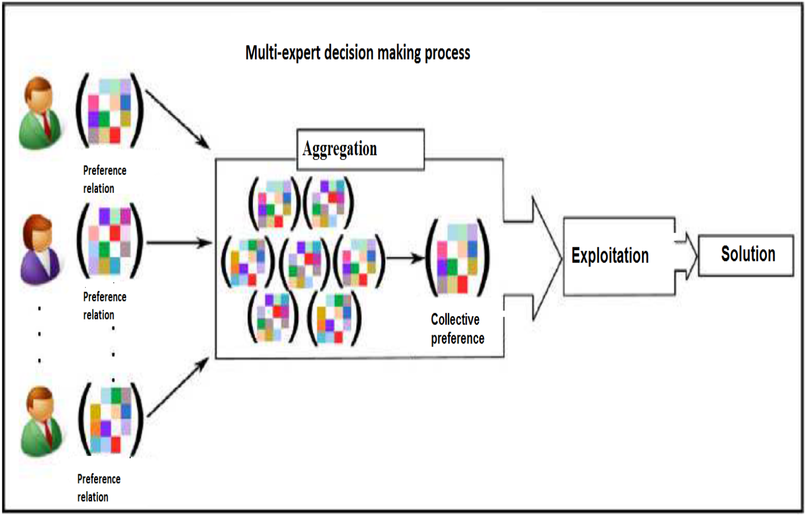
Image Reduction Using Means on Discrete Product Lattices. G. Beliakov, H. Bustince, D. Paternain IEEE Transactions on Image Processing 21 (3), 1070–1083 (2012).

An example

Set of aggregation functions ($q = 5$ chosen 3 by 3): $\wedge, \vee$, geometric mean, arithmetic mean, median



The multi-expert decision making case



Some difficulties

- Penalty functions require strong analytical conditions (quasi-convexity).
- This can be softened if we use, for instance, moderate deviation functions.
- But...

HOW DO WE CHOOSE THE CORRECT ONE?

Moderate deviation and restricted equivalence functions for measuring similarity between data. A.H. Altalhi, J.I. Forcen, M. Pagola, E. Barrenechea, H. Bustince, Z. Takác Information Sciences, 501, 19–29 (2019)

Index

A different problem

The monotonicity problem

We are asking for **monotonicity**

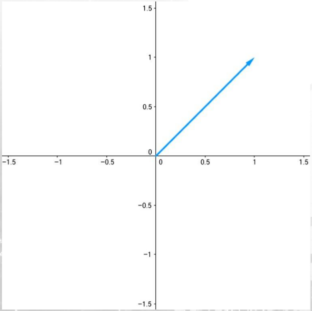
But some fusion methods are not monotone:

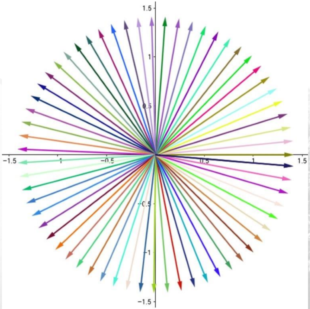
- Statistical operators (the mode)
- Implication functions
- Similarity measures
- Distances

So then?

One step ahead: directional monotonicity

- Weak monotonicity along the direction $(1, \ldots, 1)$ (2015, T. Wilkin, G. Beliakov)
- Generalization: Let's consider any direction $\vec{r} \in \mathbb{R}^n$





The formal definition

Definition

Let $\vec{r}$ be a real vector ($\vec{r} \neq 0$). A fusion function $F : [0,1]^n \to [0,1]$ is $\vec{r}$-increasing if for every $\mathbf{x} \in [0,1]^n$ and for every $c > 0$ such that $\mathbf{x} + c\vec{r} \in [0,1]^n$ it holds that:

$$F(\mathbf{x} + c\vec{r}) \geq F(x)$$

Some examples:

- Every implication function $I : [0,1]^2 \to [0,1]$ is $(-1,1)$-increasing.
- $F(x,y) = x - \max(0,(x-y)^2)$ is $(1,1)$-increasing and $(0,1)$-decreasing, but it is not $(1,0)$-increasing nor $(1,0)$-decreasing.

Directional monotonicity of fusion functions, H. Bustince, J. Fernandez, A. Kolesárová, R. Mesiar, European Journal of Operational Research 244 (1), 300-308 (2015).

Definition

Let $F : [0,1]^n \to [0,1]$ be a fusion function and let $\vec{r} \neq \vec{0}$ be an n-dimensional vector. F is said to be ordered directionally (OD) $\vec{r}$-increasing if for any $\mathbf{x} \in [0,1]^n$, for any $c > 0$ and for any permutation $\sigma : \{1, \ldots, n\} \to \{1, \ldots, n\}$ with $x_{\sigma(1)} \geq \cdots \geq x_{\sigma(n)}$ and such that

$$1 \geq x_{\sigma(1)} + cr_1 \geq \cdots \geq x_{\sigma(n)} + cr_n \geq 0$$

it holds that

$$F(\mathbf{x} + c\vec{r}_{\sigma^{-1}}) \geq F(\mathbf{x})$$

where $\vec{r}_{\sigma^{-1}} = (r_{\sigma^{-1}(1)}, \ldots, r_{\sigma^{-1}(n)})$

- Let $F : [0,1]^n \to [0,1]$ be a constant fusion function. Then, for every vector $\vec{r} \in \mathbb{R}$, F is OD $\vec{r}$-increasing and OD $\vec{r}$-decreasing.

Some examples...

- Let $F : [0,1]^n \to [0,1]$ be a constant fusion function. Then, for every vector $\vec{r} \in \mathbb{R}$, F is OD $\vec{r}$-increasing and OD $\vec{r}$-decreasing.

- Let $p > 0$. $F(x,y) = |x-y|^p$ is OD $\vec{r}$-increasing for every vector $\vec{r} = (r_1, r_2)$ such that $r_2 \leq r_1$.

Some examples...

- Let $F : [0,1]^n \to [0,1]$ be a constant fusion function. Then, for every vector $\vec{r} \in \mathbb{R}$, F is OD $\vec{r}$-increasing and OD $\vec{r}$-decreasing.

- Let $p > 0$. $F(x,y) = |x-y|^p$ is OD $\vec{r}$-increasing for every vector $\vec{r} = (r_1, r_2)$ such that $r_2 \leq r_1$.

- $F(x,y) = \frac{1}{2}(1 + \max(x,y) - 2\min(x,y))$ is OD $(2,1)$-increasing.

Some examples...

- Let $F : [0,1]^n \to [0,1]$ be a constant fusion function. Then, for every vector $\vec{r} \in \mathbb{R}$, F is OD $\vec{r}$-increasing and OD $\vec{r}$-decreasing.

- Let $p > 0$. $F(x,y) = |x-y|^p$ is OD $\vec{r}$-increasing for every vector $\vec{r} = (r_1, r_2)$ such that $r_2 \leq r_1$.

- $F(x,y) = \frac{1}{2}(1 + \max(x,y) - 2\min(x,y))$ is OD $(2,1)$-increasing.

- $F(x,y) = 1 - \max(x,y) + \frac{1}{2}\min(x,y)$ is OD $(1,2)$-increasing.

What for?

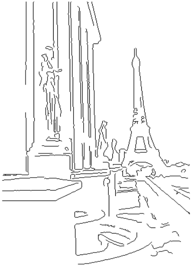
OD monotone functions are useful in problems where the relative size of inputs is relevant.

Example: Edge detection.

Ordered directionally monotone functions. Justification and application. H. Bustince, E. Barrenechea, M. Sesma-Sara, J. Lafuente, G.P. Dimuro, R. Mesiar, A. Kolesárová, IEEE Transactions on Fuzzy Systems. 26 (4), 2237-2250 (2018)

Ordered directional monotonicity in the construction of edge detectors. C. Marco-Detchart, H. Bustince, J. Fernandez, R. Mesiar, J. Lafuente, E. Barrenechea, J.M. Pintor, submitted to Fuzzy Sets and Systems.

Motivation



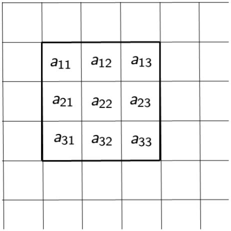
(a)Original image (b) Edge image

Specific detectors for each type of image
What is an edge?

Definition

Big enough jump between the intensity of a pixel and those of its neighbours

Application to edge detection



We calculate the differences between the central value and each 8-neighbour

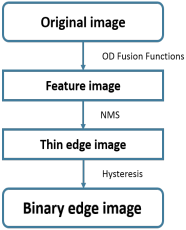
$$x_1 = |a_{22} - a_{11}|, x_2 = |a_{22} - a_{12}|, x_3 = |a_{22} - a_{13}|, x_4 = |a_{22} - a_{23}|,$$

$$x_5 = |a_{22} - a_{33}|, x_6 = |a_{22} - a_{32}|, x_7 = |a_{22} - a_{31}|, x_8 = |a_{22} - a_{21}|.$$

$$x_{\sigma(1)} \geq x_{\sigma(2)} \geq x_{\sigma(3)} \geq x_{\sigma(4)} \geq x_{\sigma(5)} \geq x_{\sigma(6)} \geq x_{\sigma(7)} \geq x_{\sigma(8)}.$$

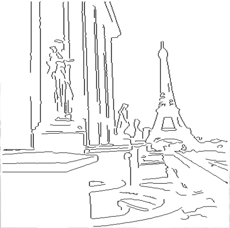
(a) Original

(b) Fuzzy edge image



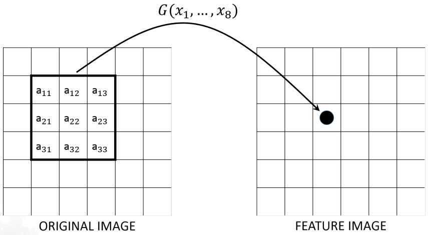
Block diagram

(c) NMS



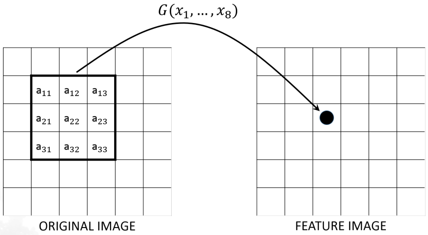
(d) Binary edge image

Figure: Sequence of the proposed edge detector



where G is an aggregation function applied to the intensity jumps:

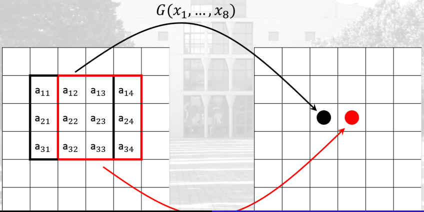
$$x_1 = |a_{22} - a_{11}|, x_2 = |a_{22} - a_{12}|, x_3 = |a_{22} - a_{13}|, x_4 = |a_{22} - a_{23}|,$$
$$x_5 = |a_{22} - a_{33}|, x_6 = |a_{22} - a_{32}|, x_7 = |a_{22} - a_{31}|, x_8 = |a_{22} - a_{21}|.$$



where G is an aggregation function applied to the intensity jumps:

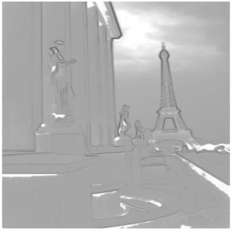
$$x_1 = |a_{22} - a_{11}|, x_2 = |a_{22} - a_{12}|, x_3 = |a_{22} - a_{13}|, x_4 = |a_{22} - a_{23}|,$$

$$x_5 = |a_{22} - a_{33}|, x_6 = |a_{22} - a_{32}|, x_7 = |a_{22} - a_{31}|, x_8 = |a_{22} - a_{21}|.$$



Motivation

(a) Fuzzy Morphology (Max-Min)



(b) Fuzzy Morphology (Schweizer-Sklar)

(c) Gravitational (S_M)

(d) OD

Figure: Gradient images obtained with different edge detectors.

Motivation

- The performance has been done using 100 images of the test subset of the Berkeley Segmentation Dataset (BSDS500)[1]
- The dataset comprises original images and ground truth images.
- Once the fuzzy edge image is obtained, we apply a thinning algorithm as Non-Maxima Suppression (NMS) (proposed by Canny, performs the suppression of all values along the line of the gradient that are not peak values.[2])
- After that, the non-supervised algorithm of hysteresis is performed to binarize the image.[3]

- D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proc. of the 8th International Conference on Computer Vision, 2001, vol. 2, pp. 416–423.
- NMS has been performed using P. Kovesis' implementation in MATLAB. http://www.csse.uwa.edu.au/pk/research/matlabfns/
- R. Medina–Carnicer, R. MuÃ±oz–Salinas, E. Yeguas–Bolivar, and L. Diaz–Mas. A novel method to look for the hysteresis thresholds for the Canny edge detector. Pattern Recognition, 44(6):1201–1211, 2011.
  P.L. Rosin, Unimodal thresholding. Pattern Recognition. 34(11), 2083-2096, 2001.

The affine construction

Proposition

Let $\mathfrak{m}$ be a fuzzy measure. Then the Choquet integral

$$C_{\mathfrak{m}}(\mathbf{x}) = \sum_{i=1}^{n} \left(x_{(i)} - x_{(i-1)}\right) \cdot \mathfrak{m}\left(A_{(i)}\right), \tag{2}$$

is OD $\vec{r}$-monotone for every non-null n-dimensional vector $\vec{r}$ such that

$$\sum_{i=1}^{n} r_i(\mathfrak{m}(A_{(i)}) - \mathfrak{m}(A_{(i+1)})) \geq 0$$

where $\mathfrak{m}(A_{(n+1)}) = 0$.

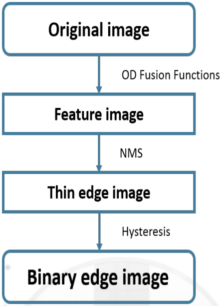
Proposition

Let $p > 0$ and let $G : [0,1]^n \to [0,1]$ be defined by

$$G(\mathbf{x}) = \begin{cases} \left(a + \sum_{i=1}^{n} b_i x_{\sigma(i)}\right)^{\frac{1}{p}} & \text{if } 0 \leq a + \sum_{i=1}^{n} b_i x_{\sigma(i)} \leq 1 \\ 0 & \text{if } a + \sum_{i=1}^{n} b_i x_{\sigma(i)} \leq 0 \\ 1 & \text{if } a + \sum_{i=1}^{n} b_i x_{\sigma(i)} \geq 1 \end{cases}$$

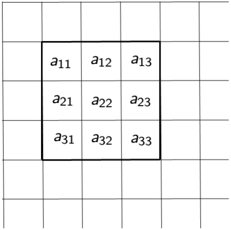
for some $a \in [0,1]$ and $b_1, \ldots, b_n \in \mathbb{R}$ such that $0 \leq a + b_1 + \cdots + b_n \leq 1$. Then G is OD $\vec{r}$-increasing for every non null vector $\vec{r}$ such that $\vec{b} \cdot \vec{r} \geq 0$.

The affine construction



- Separability Criteria for the Evaluation of Boundary Detection Benchmarks. Lopez-Molina, C.; Bustince, H.; De Baets, B. IEEE TRANSACTIONS ON IMAGE PROCESSING 25 (3) 1047-1055 (2016)
- A gravitational approach to edge detection based on triangular norms Por: Lopez-Molina, C.; Bustince, H.; Fernandez, J.; et ál.. PATTERN RECOGNITION 43 (11) 3730-3741 (2010).

Application to edge detection



We calculate the differences between the central value and each 8-neighbour

$$x_1 = |a_{22} - a_{11}|, x_2 = |a_{22} - a_{12}|, x_3 = |a_{22} - a_{13}|, x_4 = |a_{22} - a_{23}|,$$

$$x_5 = |a_{22} - a_{33}|, x_6 = |a_{22} - a_{32}|, x_7 = |a_{22} - a_{31}|, x_8 = |a_{22} - a_{21}|.$$

We order these differences in a decreasing way:

$$x_{\sigma(1)} \geq x_{\sigma(2)} \geq x_{\sigma(3)} \geq x_{\sigma(4)} \geq x_{\sigma(5)} \geq x_{\sigma(6)} \geq x_{\sigma(7)} \geq x_{\sigma(8)}.$$

CASE 1. Application to edge detection

$$\vec{r} = (x_{\sigma_{(1)}}, x_{\sigma_{(2)}}, x_{\sigma_{(3)}}, x_{\sigma_{(4)}}, x_{\sigma_{(5)}}, x_{\sigma_{(6)}}, x_{\sigma_{(7)}}, x_{\sigma_{(8)}});$$

$$\vec{b} = \begin{cases} \left(\dfrac{x_{\sigma_{(1)}}}{\sum\limits_{i=1}^{8} x_{\sigma_{(i)}}}, \ldots, \dfrac{x_{\sigma_{(7)}}}{\sum\limits_{i=1}^{8} x_{\sigma_{(i)}}}, \dfrac{x_{\sigma_{(8)}}}{\sum\limits_{i=1}^{8} x_{\sigma_{(i)}}} \right) & \text{if } \sum\limits_{i=1}^{8} x_{\sigma_{(i)}} \neq 0 \\ (0, \ldots, 0) & \text{otherwise.} \end{cases}$$

$$a = 0 \text{ and } \frac{1}{p} = 0.30.$$

CASE 2. Application to edge detection

$$\vec{r} = (x_{\sigma_{(1)}}, x_{\sigma_{(2)}}, x_{\sigma_{(3)}}, x_{\sigma_{(4)}}, x_{\sigma_{(5)}}, x_{\sigma_{(6)}}, x_{\sigma_{(7)}}, x_{\sigma_{(8)}});$$

$$\vec{b} = \left(\frac{1}{8}\left(1 - \left|x_{\sigma_{(1)}} - \operatorname*{median}_{i\in\{1,\dots,8\}}\{x_i\}\right|\right), \dots \right.$$
$$\left. \dots, \frac{1}{8}\left(1 - \left|x_{\sigma_{(8)}} - \operatorname*{median}_{i\in\{1,\dots,8\}}\{x_i\}\right|\right)\right);$$

$$a = 0 \text{ and } \frac{1}{p} = 0.30.$$

Algorithm 1 Algorithm to construct a feature image using ODM functions

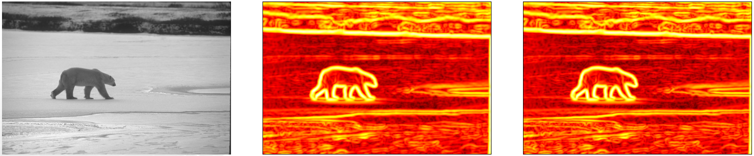
Input: A normalized greyscale image I_G and a parameter $p > 0$ to build an ODM function G as in Corollary 8.

Output: A feature image I_M.

1: **for** each pixel (x, y) of I_G **do**
2: Calculate the 8 values obtained by applying the absolute value of the difference between $\mathbb{I}_g(x, y)$ and its 8-neighbourhood;
3: Order the eight values of step 2 in a decreasing way;
4: Calculate the parameters a, $\vec{r}$ y $\vec{b}$ according to the vector obtained in step 3.
5: Build the ODM function G as in Corollary 8 with the parameters obtained in step 4.
6: Apply the ODM function G to the values obtained in step 3;
7: Assign as intensity of the pixel (x, y) of I_M the value obtained in step 6.
8: **end for**

Application to edge detection

Results obtained applying Algorithm 1 with different ODM functions.



(a) *Original* (b) *Case 1* (c) *Case 2*

Figure: Original image from BSDS [**?**] (100007) along with feature images obtained after applying Algorithm 1 with Case 1 and case 2.

In Fig. 3 we show the results obtained by applying Algorithm 1 with the two ODM functions, Case 1 and Case 2, to an original image, Fig. 3a.

Comparison of best and worst appoaches

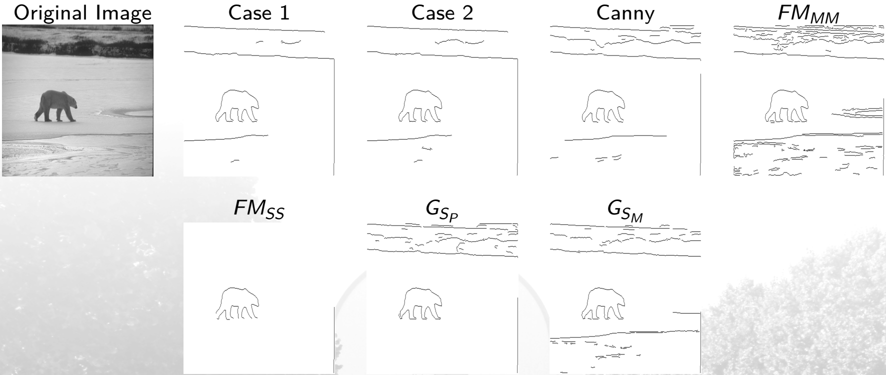
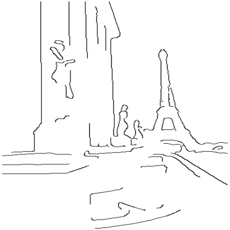


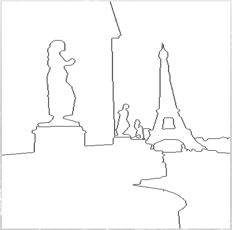
Figure: Original images and the binary images obtained with ODM functions (Case 1, Case 2), Canny, Fuzzy Morphology (FM_{SS}, FM_{MM}) and Gravitational forces (G_{S_P}, G_{S_M}), after executing steps $(S1) - (S4)$

Motivation

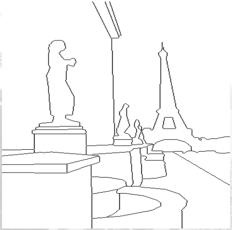
(a) Original Image



(b) Canny edge detector



(c) Human1 ground truth



(d) Human2 ground truth

(e) Human3 ground truth

Comparison measures

The following measures of *Precision* ($PREC$), *Recall* (REC) and *F-measure* (F_α) are calculated from the confusion matrix:

$$PREC = \frac{TP}{TP + FP},$$

$$REC = \frac{TP}{TP + FN},$$

$$F_\alpha = \frac{PREC \cdot REC}{\alpha \cdot PREC + (1 - \alpha) \cdot REC}.$$

We consider the commonly used $\alpha = 0.5$; in that case, $F_{0.5}$ is the harmonic mean of $PREC$ and REC.

Edge Detection Method	*PREC*	*REC*	$F_{0.5}$
$C1$	0.579	**0.794**	0.653
$C2$	0.602	0.765	0.654
FM_{SS}	0.572	0.719	0.615
C	0.687	0.618	0.631
G_{S_P}	0.649	0.649	0.650
G_{S_M}	0.661	0.665	0.641
SF	**0.753**	0.645	**0.682**

Table: Comparison of ODM functions approach with other edge detection methods as Gravitational, Fuzzy Morphology, Structured Forest and Canny in terms of *PREC*, *REC* and $F_{0.5}$.

Comparison of best and worst appoaches

	Edge Detection Methods					
	$C1$	FM_{SS}	SF	C	G_{S_P}	G_{S_M}
Best	44	14	86	6	29	21
Worst	17	86	23	39	16	19
	$C2$	FM_{SS}	SF	C	G_{S_P}	G_{S_M}
Best	50	16	86	6	27	15
Worst	9	89	27	39	20	16

Table: Comparison of best and worst approaches for 200 images of (BSDS500) in terms of $F_{0.5}$.

Fusing feature images

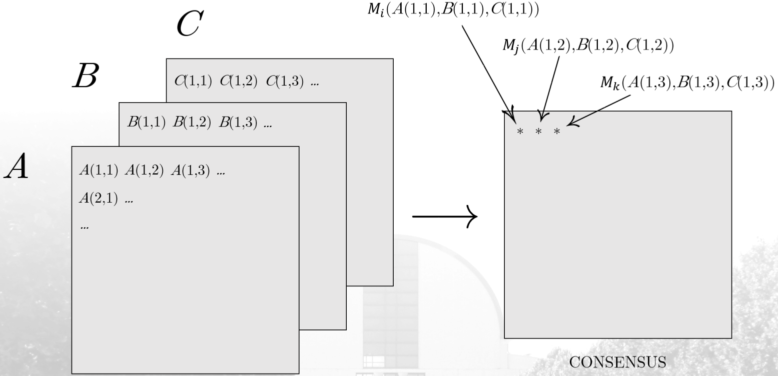


Figure: Penalty description

Consensus algorithm

Algorithm 2 Algorithm to construct a consensus feature image

Input: n feature images I_m with $m \in \{1, \ldots, n\}$.

Output: A consensus feature image $\mathfrak{O}$.

1: Calculate the value of the intensity of each pixel in the consensus feature image $\mathfrak{O}$ as the arithmetic mean of the pixels intensities in the same position of the feature images I_m with $m \in \{1, \ldots, n\}$.

Some results

Edge Detection Method	*PREC*	*REC*	$F_{0.5}$
C1-C2	0.586	**0.785**	0.654
C1-Canny	0.661	0.682	0.652
C1-FM_{SS}	0.582	0.784	0.651
C1-G_{S_P}	0.620	0.749	0.660
C1-G_{S_M}	0.625	0.731	0.654
C1-SF	0.715	0.724	**0.705**
C2-Canny	0.668	0.669	0.650
C2-FM_{SS}	0.602	0.754	0.649
C2-G_{S_P}	0.628	0.731	0.657
C2-G_{S_M}	0.635	0.715	0.653
C2-SF	0.720	0.710	0.701
Canny-FM_{SS}	0.675	0.650	0.644
Canny-G_{S_P}	0.677	0.666	0.651
Canny-G_{S_M}	0.674	0.648	0.641
Canny-SF	**0.728**	0.671	0.683
FM_{SS}-G_{S_P}	0.624	0.734	0.655
FM_{SS}-G_{S_M}	0.631	0.711	0.649
FM_{SS}-SF	0.722	0.708	0.701
G_{S_P}-G_{S_M}	0.654	0.687	0.650
G_{S_P}-SF	0.722	0.688	0.687
G_{S_M}-SF	0.725	0.673	0.681

	Edge Detection Methods											
	*		FM_{SS}		C		G_{S_P}		G_{S_M}		SF	
	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗
C1-C2	42	**11**	16	88	6	39	31	20	18	16	**87**	26
C1-Canny	18	**2**	26	94	6	39	41	21	18	16	**91**	28
C1-FM_{SS}	43	**15**	11	87	6	39	34	19	20	16	**86**	24
C1-G_{S_P}	43	**6**	22	93	7	39	18	18	19	16	**91**	28
C1-G_{S_M}	30	**4**	22	93	6	39	33	21	17	15	**92**	28
C1-SF	**97**	**1**	20	96	4	39	26	21	15	16	38	27
C2-Canny	15	**1**	28	95	6	39	42	21	18	16	**91**	28
C2-FM_{SS}	38	**11**	14	88	6	39	37	19	17	15	**88**	28
C2-G_{S_P}	38	**3**	25	95	7	39	22	19	18	16	**90**	28
C2-G_{S_M}	26	**2**	24	94	6	39	36	21	15	16	**93**	28
C2-SF	**96**	**1**	20	96	4	39	29	21	15	16	36	27
Canny-FM_{SS}	16	**1**	29	96	3	38	42	21	20	16	**90**	28
Canny-G_{S_P}	15	**3**	28	96	7	38	37	20	19	15	**94**	28
Canny-G_{S_M}	9	**5**	28	96	7	37	43	21	19	13	**94**	28
Canny-SF	52	**2**	27	95	5	39	36	20	16	16	**64**	28
FM_{SS}-G_{S_P}	42	**9**	23	91	6	38	19	18	19	16	**91**	28
FM_{SS}-G_{S_M}	24	**4**	22	93	6	39	39	21	17	15	**92**	28
FM_{SS}-SF	**98**	**1**	20	96	4	39	28	21	16	16	34	27
G_{S_P}-G_{S_M}	18	**6**	26	95	7	38	37	19	17	14	**95**	28
G_{S_P}-SF	61	**6**	28	95	5	38	28	18	16	16	**62**	27
G_{S_M}-SF	48	**3**	29	95	5	39	36	20	15	16	**67**	27

Edge detection using penalty functions

Figure: Original images (first row) and the binary images obtained with penalty functions Case 1 (second row) and Case 2 (third row)

A reference

Ordered directional monotonicity in the construction of edge detectors.
C. Marco-Detchart, H. Bustince, R. Mesiar, J. Lafuente, E. Barrenechea, J.M. Pintor
Fuzzy Sets and Systems, in press

Definition

An aggregation function is a function $M : [0,1]^n \to [0,1]$ such that:

1. *M is increasing;*
2. $M(0, \ldots, 0) = 0$
3. $M(1, \ldots, 1) = 1$.

Definition

A function $F : [0,1]^n \to [0,1]$ *is said to be an n-ary pre-aggregation function if the following conditions hold:*

(PA1) *There exists a real vector* $\vec{r} \in [0,1]^n$ ($\vec{r} \neq \vec{0}$) *such that* F *is* $\vec{r}$*-increasing.*

(PA2) F *satisfies the boundary conditions:* $F(0,\ldots,0) = 0$ *and* $F(1,\ldots,1) = 1$.

If F is a pre-aggregation function with respect to a vector $\vec{r}$ we just say that F is an $\vec{r}$-pre-aggregation function.

Preaggregation Functions: Construction and an Application. Giancarlo Lucca; José Antonio Sanz; Gracaliz Pereira Dimuro; Benjamín Bedregal; Radko Mesiar; Anna Kolesárová; Humberto Bustince, IEEE Transactions on Fuzzy Systems, 24(2), 260 - 272 (2016).

Pre-aggregation functions

(i) The mode, $Mod(x_1, \ldots, x_n)$ is $(1, \ldots, 1)$-increasing, and it is a particular case of pre-aggregation function which is not an aggregation function.

Pre-aggregation functions

(i) The mode, $Mod(x_1, \ldots, x_n)$ is $(1, \ldots, 1)$-increasing, and it is a particular case of pre-aggregation function which is not an aggregation function.

(ii) $F(x, y) = x - (\max\{0, x - y\})^2$ is, for instance, $(0, 1)$-increasing, and it is an example of a pre-aggregation function which is not an aggregation function.

Pre-aggregation functions

(i) The mode, $Mod(x_1, \ldots, x_n)$ is $(1, \ldots, 1)$-increasing, and it is a particular case of pre-aggregation function which is not an aggregation function.

(ii) $F(x,y) = x - (\max\{0, x-y\})^2$ is, for instance, $(0,1)$-increasing, and it is an example of a pre-aggregation function which is not an aggregation function.

(iii) Take $\lambda \in]0,1[$. The weighted Lehmer mean $L_\lambda : [0,1]^2 \to [0,1]$, given by

$$L_\lambda(x,y) = \frac{\lambda x^2 + (1-\lambda)y^2}{\lambda x + (1-\lambda)y}$$

(with convention $0/0 = 0$) is $(1-\lambda, \lambda)$-increasing, so it is a pre-aggregation function.

The key concept: An important remark

Theorem

If $A : [0,1]^n \to [0,1]$ *is an aggregation function, then* A *is also a pre-aggregation function.*

On some classes of directionally monotone functions. H. Bustince, R. Mesiar, A. Kolesárová, G.P. Dimuro, J. Fernandez, I. Diaz, S. Montes, Fuzzy Sets and Systems, in press.

How do we build pre-aggregation functions

- The idea is to modify some well-known aggregation functions.

- We arrive at the method from a specific problem.

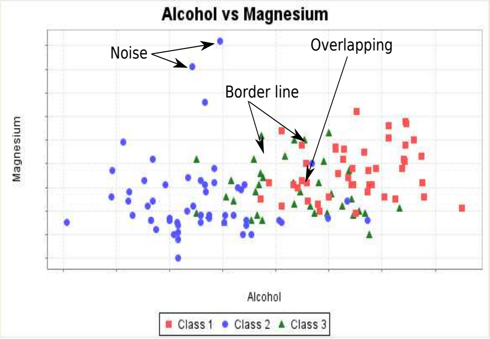
A classification problem

- Classification problem:

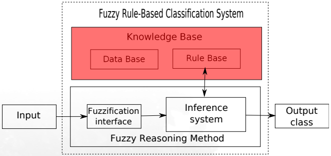
  P examples: $X = \{x_1, ..., x_p\}$
  n attributes: $\mathcal{A} = \{a_1, ..., a_n\}$
  M classes: $C = \{c_1, ..., c_M\}$

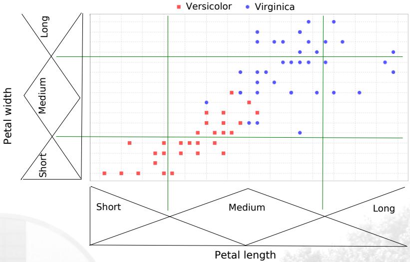
Classifier:

$$D : \mathcal{A} \to C$$



A classification problem



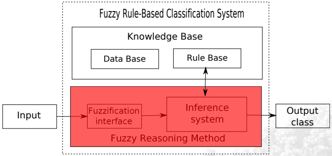


R_1: If *Width* is *Short* then *Class* = *Versicolor*
R_2: If *Length* is *Long* then *Class* = *Virginica*
R_3: If *Width* is *Long* then *Class* = *Virginica*
R_4: If *Length* is *Average* and *Width* is *Average* then *Class* = *Virginica*

A classification problem

$$R_j: \text{ If } x_{p1} \text{ is } A_{j1} \text{ and } \ldots \text{ and } x_{pn} \text{ is } A_{jn} \text{then } Class = C_j \text{ with } RW_j$$

- Fuzzy Reasoning Method:
  1. *Matching degree:*
     $\mu_{A_j}(x_p) = T(\mu_{A_{j1}}(x_{p1}), \ldots, \mu_{A_{jn}}(x_{pn}))$
  2. *Association degree:*
     $$\mathrm{b}_j^k = h(\mu_{A_j}(x_p), RW_j^k)$$
  3. *Association degree by classes:*
     $$Y_k = f(b_j^k,\ b_j^k > 0)$$
  4. *Classification:*
     $$C_{best} = \arg \max_{k=1,\cdots,M} (Y_k)$$



- k = 1, …, M (n. classes).
- j = 1,…, L (n. rules).

O. Cordón, M. J. del Jesús, F. Herrera: A proposal on reasoning methods in fuzzy rule-based classification systems. Int. J. Approx. Reason., 20:1 (1999) 21–45.

- In step 3, usually the maximum is used...
- ... so we are ignoring information provided by almost every rule!!!
- What happens if we try to take into account this information?
- Let's use, for instance, Choquet integral...

Experimental setting

1. 27 datasets selected form the KEEL repository
2. 5 folder cross validation method
3. *Accuracy rate* to measure the performance
4. Fuzzy classifier: FARC-HD
   - Conjunction operator: product T-norm
   - Rule weight: certainty factor
   - 5 linguistic labels per variable
   - Minimum support: 0.05
   - Maximum confidence: 0.8
   - Maximum tree depth: 3
5. Statistical study
   1. Multiple comparisons: Friedman's Aligned ranks + Holm
   2. Pairwise comparisons: Wilcoxon

J. Alcala-Fdez, R. Alcala, F. Herrera, A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning, IEEE Transactions on Fuzzy Systems 19 (5) (2011) 857–872.

A first lucky idea

So we use the Choquet integral... with a "small" change:

The first idea

$$\begin{aligned}
C_{\mathfrak{m}}(\mathbf{x}) &= \sum_{i=1}^{n} \left(x_{(i)} - x_{(i-1)}\right) \cdot \mathfrak{m}\left(A_{(i)}\right) \\
\Downarrow & \qquad\qquad\qquad\qquad \Downarrow \\
C_{\mathfrak{m}}^{M}(\mathbf{x}) &= \sum_{i=1}^{n} M\left(x_{(i)} - x_{(i-1)}, \mathfrak{m}\left(A_{(i)}\right)\right)
\end{aligned}$$

A first lucky idea

Theorem

Let $M : [0,1]^2 \to [0,1]$ *be a function such that for all* $x, y \in [0,1]$ *it satisfies* $M(x,y) \leq x$, $M(x,1) = x$, $M(0,y) = 0$ *and* M *is (1,0)-increasing. Then, for any fuzzy measure* $\mathfrak{m}$, $C_{\mathfrak{m}}^{M}$ *is a pre-aggregation function which is idempotent and averaging, i.e.,*

$$\min(x_1, \ldots, x_n) \leq C_{\mathfrak{m}}^{M}(x_1, \ldots, x_n) \leq \max(x_1, \ldots, x_n).$$

- Results of the 20 pre-aggregations considered

- Testing results

	Uniform	Dirac	Wmean	OWA	Power_GA
Product	78.68 (7)	**78.01 (3)**	78.12 (4)	77.33 (4)	78.55 (5)
Minimum	78.85 (7)	77.81 (0)	78.75 (7)	78.33 (10)	79.00 (7)
Łukasiewicz	76.61 (1)	77.81 (1)	76.92 (0)	76.44 (1)	78.14 (0)
Drastic	76.66 (0)	77.81 (0)	76.66 (1)	76.66 (2)	76.66 (1)
Nilpotent	76.88 (1)	77.81 (0)	76.76 (3)	76.60 (1)	78.78 (5)
Hamacher	**79.16 (8)**	77.81 (1)	**79.19 (10)**	**78.61 (7)**	**79.42 (7)**

- Statistical study

Aligned Friedman (APV Holm)

	Uniform	Dirac	WMean	OWA	Power_GA
Product	42.94 (0.21)	**38.13**	51.09 (0.002)	53.91 (0.003)	50.78 (0.004)
Minimum	45.13 (0.21)	43.38 (0.771)	42.13 (0.054)	35.24 (0.828)	41.20 (0.112)
Hamacher	**50.22**	41.18 (0.771)	**29.78**	**33.85**	**31.02**

Pre-aggregation Functions: Construction and an Application. G. Lucca, J. Sanz, G. Dimuro, B. Bedregal, R. Mesiar, A. Kolesárová, H. Bustince, IEEE Transactions on Fuzzy Systems 24 (2) 260–272 (2016).

Experimental results

- Comparison of the best pre-aggregation versus the winning rule FRM (maximum)

- Testing results

Dataset	WR	Power_GA+Ham
App	**84.89**	82.99
Bal	82.08	**82.72**
Ban	84.30	**85.96**
Bnd	68.56	**72.13**
Bup	61.16	**65.80**
Cle	55.23	**55.58**
Eco	75.61	**80.07**
Gal	**63.11**	63.10
Hab	71.22	**72.21**
Hay	79.46	**79.49**
Iri	**94.67**	93.33
Led	**69.80**	68.60
Mag	79.60	**79.76**
New	94.42	**95.35**
Pag	**94.52**	94.34
Pho	82.01	**83.83**
Pim	**75.38**	73.44
Rin	**90.00**	88.79
Sah	67.31	**70.77**
Sat	**80.40**	**80.40**
Seg	92.99	**93.33**
Tit	**78.87**	**78.87**
Two	84.32	**85.27**
Veh	67.62	**68.20**
Win	94.36	**96.63**
Wis	96.49	**96.78**
Yea	**56.54**	56.53
Mean	78.70	**79.42**

- Statistical study: Wilcoxon

Comparison	R^+	R^-	p-value
Power_GA+Ham vs. WR	267.5	110.5	0.06

But...

If we take:

$$C_{\mathfrak{m}}^{M}(\mathbf{x}) = \sum_{i=1}^{n} M\left(x_{(i)} - x_{(i-1)}, \mathfrak{m}\left(A_{(i)}\right)\right),$$

we overcome the winning rule (the maximum).

We want more: let's go for FURIA and FARC!!!

WHAT ELSE CAN WE DO??

One step more

$$
\begin{aligned}
C_{\mathfrak{m}}(\mathbf{x}) &= \sum_{i=1}^{n} \left(x_{(i)} - x_{(i-1)}\right) \cdot \mathfrak{m}\left(A_{(i)}\right) \\
\Downarrow \qquad & \qquad\qquad\qquad\qquad\qquad \Downarrow \\
C_{\mathfrak{m}}^{M}(\mathbf{x}) &= \sum_{i=1}^{n} M\left(x_{(i)} - x_{(i-1)}, \mathfrak{m}\left(A_{(i)}\right)\right)
\end{aligned}
$$

$$
\begin{array}{rcl}
C_{\mathfrak{m}}(\mathbf{x}) & = & \sum_{i=1}^{n} \left(x_{(i)} - x_{(i-1)}\right) \cdot \mathfrak{m}\left(A_{(i)}\right) \\
\Downarrow & & \qquad\qquad\qquad \Downarrow \\
C_{\mathfrak{m}}^{M}(\mathbf{x}) & = & \sum_{i=1}^{n} M\left(x_{(i)} - x_{(i-1)}, \mathfrak{m}\left(A_{(i)}\right)\right)
\end{array}
$$

The second idea

$$
\begin{array}{rcl}
C_{\mathfrak{m}}(\mathbf{x}) & = & \sum_{i=1}^{n} \left(x_{(i)} \cdot \mathfrak{m}(A_{(i)}) - x_{(i-1)} \cdot \mathfrak{m}\left(A_{(i)}\right)\right) \\
\Downarrow & & \qquad\qquad\qquad \Downarrow \\
C_{\mathfrak{m}}^{F_1,F_2}(\mathbf{x}) & = & \sum_{i=1}^{n} F_1(x_{(i)}, \mathfrak{m}(A_{(i)})) - F_2(x_{(i-1)}, \mathfrak{m}(A_{(i)}))
\end{array}
$$

F_1-F_2based Choquet-like integrals

To get a value smaller than 1 we do:

$$C_{\mathfrak{m}}^{(F_1,F_2)}(\mathbf{x}) = \min\left\{1, \sum_{i=1}^{n} F_1\left(x_{(i)}, \mathfrak{m}\left(A_{(i)}\right)\right) - F_2\left(x_{(i-1)}, \mathfrak{m}\left(A_{(i)}\right)\right)\right\},$$

Conditions for F_1 and F_2?

Proposition *

Let $F_1, F_2 : [0,1]^2 \to [0,1]$ be two bivariate functions such that, for every $x, y \in [0,1]$, it holds that:

1. F_1 is $(1,0)$-increasing;
2. $F_1(0,x) = F_2(0,x)$;
3. $F_1(0,1) = F_2(0,1) = 0$;
4. $F_1(1,1) = 1$;
5. $F_1(x,y) \geq F_2(x,y)$.

Then, for any fuzzy measure $\mathfrak{m}$, the function $C_{\mathfrak{m}}^{(F_1,F_2)}$ is well-defined and satisfies:

$$0 \leq C_{\mathfrak{m}}^{(F_1,F_2)}(\mathbf{x}) \leq 1$$

for every $\mathbf{x} \in [0,1]^n$.

Does it work?

Proposition

If we take:

- $F_1(x, y) = \sqrt{xy}$
- $F_2(x, y) = \max(x + y - 1, 0)$,

then

$$C_{\mathfrak{m}}^{(F_1,F_2)}(\mathbf{x}) = \min\left\{1, \sum_{i=1}^{n} F_1\left(x_{(i)}, \mathfrak{m}\left(A_{(i)}\right)\right) - F_2\left(x_{(i-1)}, \mathfrak{m}\left(A_{(i)}\right)\right)\right\}$$

is a non-averaging pre-aggregation function.

Results

Table 1: Results achieved in testing considering the $F_1 F_2$ approach

Dataset	FURIA	AC	ProbSum	GM_LK
appendicitis	**87.71**	83.03	85.84	84.89
balance	83.68	85.92	87.20	**89.76**
banana	**88.57**	85.30	84.85	85.23
bands	69.40	68.28	68.82	**70.49**
bupa	**70.14**	67.25	61.74	66.67
cleveland	56.57	56.21	**59.25**	58.57
contraceptive	**54.17**	53.16	52.21	53.50
ecoli	80.06	82.15	80.95	**84.53**
glass	**72.91**	65.44	64.04	64.99
haberman	72.55	**73.18**	69.26	**73.18**
hayes-roth	**81.00**	77.95	77.95	79.43
ion	**89.75**	88.90	88.32	89.75
iris	94.00	94.00	**95.33**	94.67
led7digit	**71.80**	69.60	69.20	69.60
magic	80.65	**80.76**	80.39	80.18
newthyroid	94.88	94.88	94.42	**96.28**
pageblocks	95.25	95.07	94.52	**95.98**
penbased	92.45	92.55	**93.27**	92.64
phoneme	**85.90**	81.70	82.51	82.44
pima	**76.17**	74.74	75.91	75.26
ring	85.54	**90.95**	90.00	90.41
saheart	70.33	68.39	69.69	**70.56**
satimage	**82.27**	79.47	80.40	79.47
segment	**97.32**	93.12	92.94	92.86
shuttle	**99.68**	95.59	94.85	97.33
sonar	78.90	78.36	82.24	**83.23**
spectfheart	77.88	77.88	77.90	**80.12**
titanic	78.51	**78.87**	**78.87**	**78.87**
twonorm	88.11	90.95	90.00	**91.76**
vehicle	**70.21**	68.56	68.09	68.67
wine	93.78	**96.03**	94.92	**96.03**
wisconsin	96.63	96.63	**97.22**	96.34
yeast	58.22	58.96	**59.03**	58.96
Mean	**81.06**	80.12	80.07	80.99

Generalizations of the Choquet integral

- *Improving the performance of fuzzy rule-based classification systems based on a non-averaging generalization of CC-integrals named C_{F1F2}-integrals*, Giancarlo Lucca; Gracaliz Pereira Dimuro; Javier Fernandez; Humberto Bustince; Benjamín Bedregal; José Antonio Sanz. IEEE Transactions on Fuzzy Systems,27, 124–134 (2018).
- *The state-of-art of the generalizations of the Choquet integral: From aggregation and pre-aggregation to ordered directionally monotone functions*, Graa liz Pereira Dimuro, Javier Fernandez, Benjamin Bedregal, Radko Mesiar, Jose Antonio Sanz, Giancarlo Lucca, Humberto Bustince, Information Fusion, 57, 27–43, (2020).
- *A generalization of the Choquet integral defined in terms of the Möbius transform*, Javier Fernandez, Humberto Bustince, Lubomira Horánska, Radko Mesiar, Andrea Stupnánová. IEEE Transactions on Fuzzy Systems, in press.

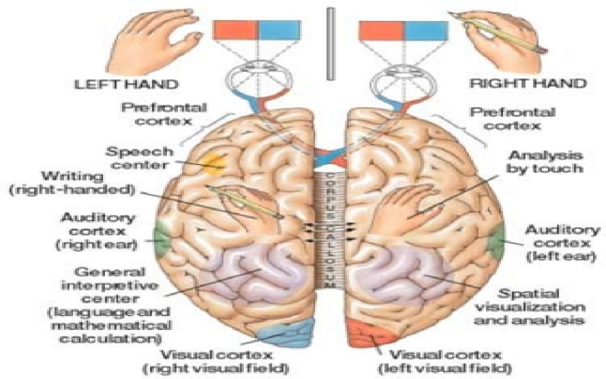
Index

We have a set of possible functions to fuse data, which works very well in some specific problems

Can we find another types of problems where it can be useful?

The case of the computational brain

Consider the problem of determining whether a subject is thinking of moving the left or the right hand.



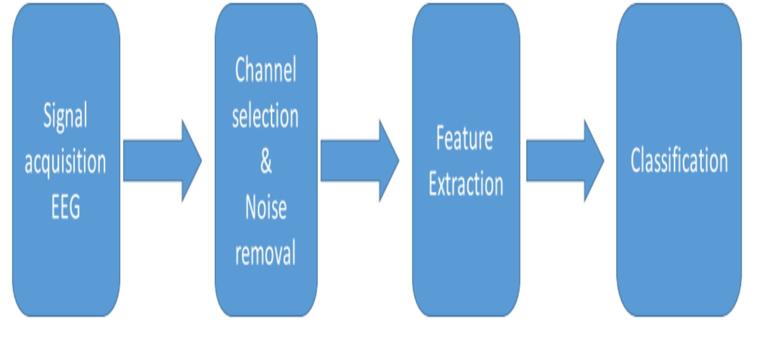
The case of the computational brain

Consider the problem of determining whether a subject is thinking of moving the left or the right hand.

Classification problem with two classes

Not appropriate for deep learning!

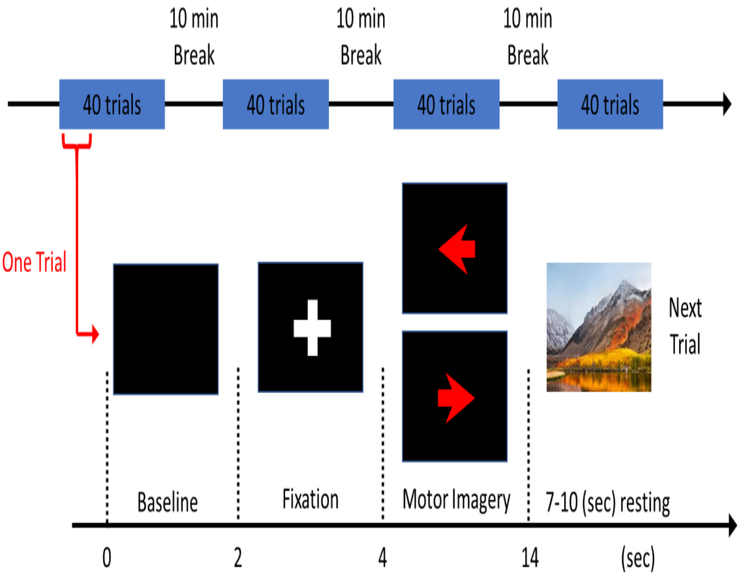
How can we do it



How can we do it

Subject no.	left-hand movement imagery trials	right-hand movement imagery trials
1	85	75
2	85	75
3	82	78
4	83	77
5	74	86

How can we do it



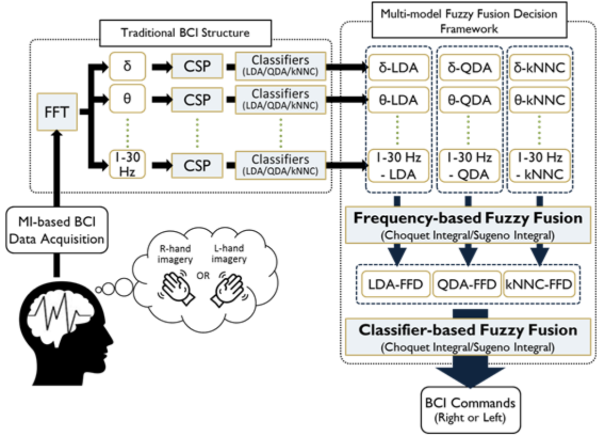
Preprocessing

- 32 EEG signals are collected.
- The signals are preprocessed with FFT to get features in five bands.
- Common spatial pattern is applied to get well-separated sub-components.

We use three classifiers:

- Lineal discriminant analysis.
- Quadratic discriminant analysis.
- kNN (with $k = 9$)

STRUCTURE OF THE ALGORITHM:



Multimodal Fuzzy Fusion for Enhancing the Motor-Imagery-based Brain Computer Interface, Li-Wei Ko, Yi-Chen Lu, Humberto Bustince, Yu-Cheng Chang, Yang Chang, Javier Fernandez, Yu-Kai Wang, Jose Antonio Sanz, Gracaliz Pereira Dimuro, Chin-Teng Lin, IEEE Computational Intelligence Magazine.14 (1), 96–106 (2019)

STRUCTURE OF THE ALGORITHM:

We make two steps:

1. Fuse the results for each band and each classifier.
2. Fuse the global result of each classifier.

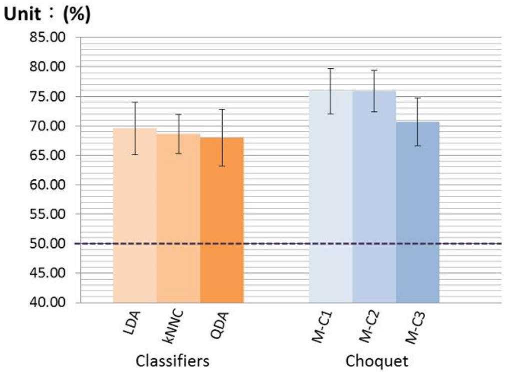
We use aggregation and pre-aggregation functions to fuse the results of each classifier

- M-C1: Choquet.
- M-C2: C_F integral with F the Hamacher t-norm:

$$F(x,y) = \begin{cases} 0 & \text{if } x = y = 0 \\ \frac{xy}{x+y-xy} & \text{otherwise.} \end{cases}$$

- M-C3: C_{F1F2} with $F1 = F2 = \min$.

Comparison of the individual classifiers



Comparison of the individual classifiers

What else can we do?

Sugeno-based construction method

Discrete Sugeno integral $S_{\mathfrak{m}} \colon [0,1]^n \to [0,1]$ can be written as

$$S_{\mathfrak{m}}(\mathbf{x}) = \bigvee_{i=1}^{n} \min \left\{ x_{(i)}, \mathfrak{m}\left(A_{(i)}\right) \right\}.$$

What happens if we replace the minimum by another aggregation function?

Sugeno-based construction method

Discrete Sugeno integral $S_{\mathfrak{m}}\colon [0,1]^n \to [0,1]$ can be written as

$$S_{\mathfrak{m}}(\mathbf{x}) = \bigvee_{i=1}^{n} \min\left\{x_{(i)}, \mathfrak{m}\left(A_{(i)}\right)\right\}.$$

What happens if we replace the minimum by another aggregation function?

$$S_{\mathfrak{m}}^{M}(\mathbf{x}) = \bigvee_{i=1}^{n} M\left(x_{(i)}, \mathfrak{m}\left(A_{(i)}\right)\right). \tag{3}$$

Sugeno-based construction method

Proposition

Let $M\colon [0,1]^2 \to [0,1]$ *be a function increasing in the first variable and let for each* $y \in [0,1]$, $M(0,y) = 0$ *and* $M(1,1) = 1$*. Then* $S_{\mathfrak{m}}^{M}$ *defined in (3) is a pre-aggregation function for any fuzzy measure* $\mathfrak{m}$*.*

Pre-aggregations vs. BCI

We use aggregation and pre-aggregation functions to fuse the results of each classifier

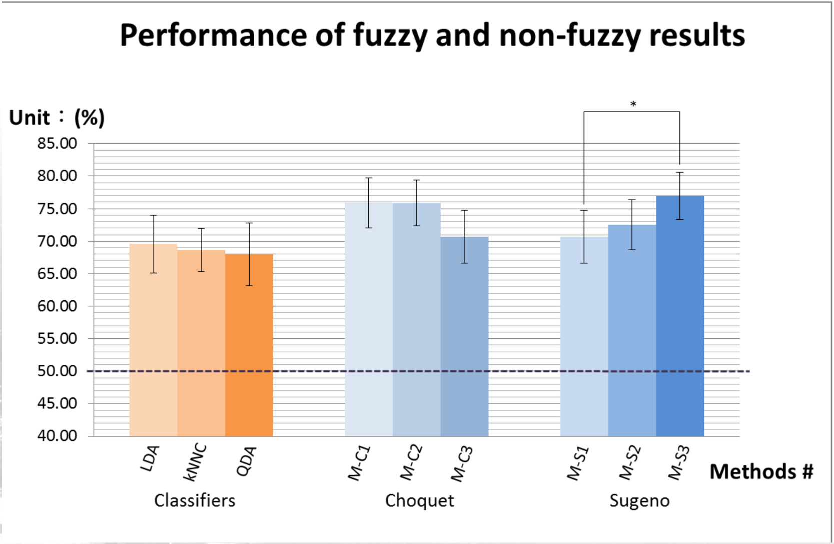
- M-S1: Sugeno.
- M-S2: S^M integral with M the Hamacher t-norm:

$$F(x,y) = \begin{cases} 0 & \text{if } x = y = 0 \\ \frac{xy}{x+y-xy} & \text{otherwise.} \end{cases}$$

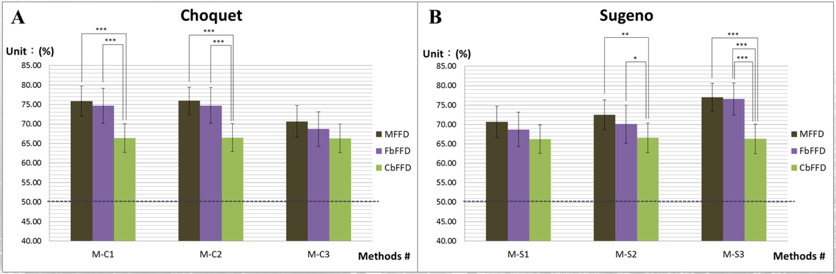
- M-S3: S^M integral with M given by:

$$M(x,y) = x|2y-1|$$

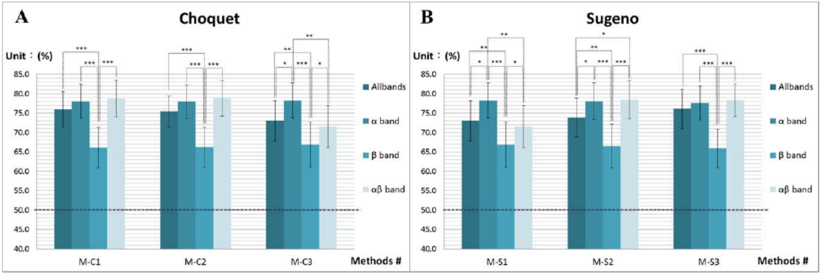
Comparison of results:



Comparison of results:



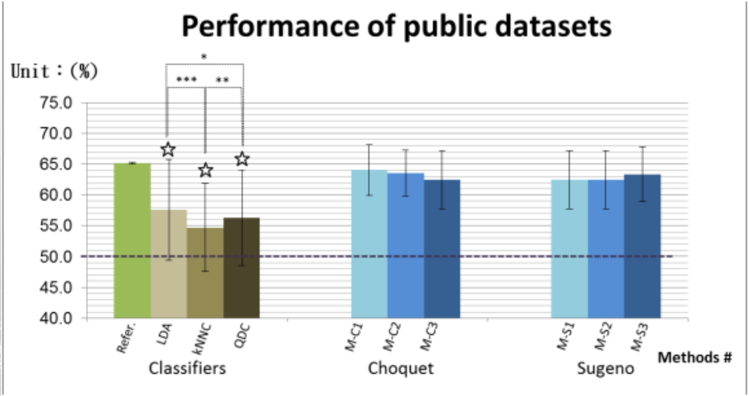
Comparison of results:



Multimodal Fuzzy Fusion for Enhancing the Motor-Imagery-based Brain Computer Interface, Li-Wei Ko, Yi-Chen Lu, Humberto Bustince, Yu-Cheng Chang, Yang Chang, Javier Fernandez, Yu-Kai Wang, Jose Antonio Sanz, Gracaliz Pereira Dimuro, Chin-Teng Lin, IEEE Computational Intelligence Magazine,14 (1), 96–106 (2019)

And for a harder problem?

- Benchmark with 4 classes: left hand, right hand, foot, tongue.
- 288 trials, 72 per class
- 22 channels per signal.



One video

Index

Conclusions

- We have explained possible ways of generalizing aggregation functions.
- We have discussed in particular how these functions can be obtained in terms of Choquet and Sugeno integrals.
- We have seen the applicability of our results in classification problems, in particular in the computational brain.

Conclusions: The computational brain

- Data fusion based on Choquet and Sugeno integrals improves the results.
- Only with the band α, accuracy is higher than 80
- This improves applicability for patients with communication difficulties.

Many thanks!!!!