Mejoras a la Capacidad de Generalización de la Inteligencia Artificial

Javier Fumanal Idocin

Estadística, Informática, Matemáticas

Universidad Pública de Navarra

A thesis submitted for the degree of

PhilosophiæDoctor (PhD)

2023, Mayo

https://doi.org/10.48035/Tesis/2454/45885

Abstract

Information fusion is a crucial aspect of modern data analysis and decision making. It involves the integration of multiple sources of information in order to form a more complete and accurate understanding of a given subject. This process is particularly important in fields such as computer science, engineering, and natural sciences, where large amounts of data are generated from a variety of sources and must be synthesised to make informed decisions. Information fusion is also essential in the design and implementation of intelligent systems, as it allows the integration of various sensors and data sources to make more accurate predictions and recommendations.

From a mathematical point of view, one way to study this problem is through the idea of fusion functions, which take as input a vector of numbers and return a single one, representative of them. A relevant kind of fusion function is the family of aggregation functions. These functions hold two boundary conditions and monotonicity with respect to the inputs, which induce some desirable properties to the function output. However, information fusion in applied systems comprises more than this theoretical notion. As the heterogeneity, the structure, and the volume of the data become more relevant, other approaches to tackle this problem have arisen. For example, in a network structure, the different inputs are associated among each other according to a pre-established set of relationships; in time series, data present temporal dependencies. When dealing with non-structured data, like text, audio, and image, deep learning approaches have been very successful in transforming this kind of data into vectorial representations of real numbers using series of affine transformations.

Despite previous efforts in the field, the problem of effectively combining diverse and heterogeneous sources of information, remains an open and active area of research. This is due to the challenges inherent in integrating multiple sources that

may be in different formats and may have conflicting or incomplete information. For example, how the information measured relates to other sources of data and how reliable those measures are is highly dependent on the measurement procedure. Indeed, systems that fuse the information from those different sources shall present additional complexities as well when taking into account the particularities of each feature considered.

In this dissertation, we propose a collection of functions and algorithms to take into account possible interactions, heterogeneities, and uncertainties when working with different sources of information. We do so by means of aggregation theory and social network analysis, and we focus especially on those cases where deep learning approaches are not so successful. We apply these results to a wide range of problems, including the classification of brain computer interface signals, the classification of standard tabular data, the detection of anomalies, and the detection of communities in social networks.

Structure of the dissertation

This dissertation is divided into two parts:

- Part I: we introduce the different perspectives of information fusion. We also present our theoretical proposals for aggregation theory and social network analysis. Finally, we motivate our design choices in BCI signal classification and introduce the algorithms developed in this dissertation for community detection, classification, and anomaly detection.

- Part II: we present a collection of the 9 papers published, accepted, and submitted related to this dissertation.

Part II is made up of these publications.

1. Fumanal-Idocin J., Wang Y. -K., Lin C. -T., Fernández J. , Sanz J. A. and Bustince H., ”Motor-Imagery-Based Brain–Computer Interface Using Signal Derivation and Aggregation Functions,” *IEEE Transactions on Cybernetics*, 52(8), 7944-7955.

2. Fumanal-Idocin, J., Takáč, Z., Fernández, J., Sanz, J. A., Goyena, H., Lin, C. T., Wang, Y.K., & Bustince, H. (2021). Interval-Valued Aggregation Functions Based on Moderate Deviations Applied to Motor-Imagery-Based Brain–Computer Interface. *IEEE Transactions on Fuzzy Systems*, 30(7), 2706-2720.

3. Fumanal-Idocin J., Vidaurre C., Fernandez J., Gomez M., Andreu-Perez J., Prasad M. & Bustince H. (2023) Supervised Penalty-based Aggregation Applied to Motor-Imagery based Brain-Computer-Interface *Pattern Recognition*. (submitted)

4. Fumanal-Idocin, J., Rodriguez-Martinez, I., Indurain, A., Minárová, M., & Bustince, H. (2022). A Generalization of the Aggregation of Forces in the Gravitational Clustering to Perform Anomaly Detection. *Information Sciences*.

5. Fumanal-Idocin, J., Alonso-Betanzos, A., Cordón, O., Bustince, H., & Minárová, M. (2020). Community detection and social network analysis based on the Italian wars of the 15th century. *Future Generation Computer Systems*, 113, 25-40.

6. Fumanal-Idocin, J., Takáč, Z., Horanská, Ľ., da Cruz Asmus, T., Dimuro, G., Vidaurre, C. Vidaurre, J. Fernandez & Bustince, H. (2022). A generalization of the Sugeno integral

to aggregate interval-valued data: An application to brain computer interface and social network analysis. Fuzzy Sets and Systems, 451, 320-341.

7. Fumanal-Idocin, J., Cordón O., Dimuro G., Lopez-de-Hierro A.F. & Bustince, H. (2022).Quantifying External Information in Social Network Analysis: an Application to Comparative Mythology. *IEEE Transactions on Cybernetics*.

8. Fumanal-Idocin J., Cordón O. & Bustince H. (2023) The Krypteia ensemble: designing classifier ensembles using an ancient Spartan military tradition *Information Fusion*.

9. Fumanal-Idocin J., Andreu-Perez J. Cordón O., Hagras H., & Bustince H. (2023) Enhancing and explaining feature extraction in artistic image classification *Information Fusion* (submitted).

Acknowledgements

La verdadera amistad no se mide por
intereses temporales, sino que se bebe
por amor gratuito
San Agustín de Hipona

Quodcumque donum alicui damus,
primum donum illis damus, quod eos
diligamus.
Santo Tommaso d'Aquino

Dedico este pequeño espacio del documento a recordar algunas de las personas que han pasado por mi vida en esta tesis. Si fuese algo mas justo debería remontarme un poco mas atras, pues muchas de las que me encaminaron a emprender esta aventura (por llamarla de alguna manera). Poner fe en algo siempre tiene más merito que medirlo una vez cumplido. No obstante, encomendándome a San Judas Tadeo (el bueno), patrón de las causas perdidas, y que probablemente haya hecho algo por esta tesis también, comienzo mi ejercicio.

Muchos de los agradecimientos que aquí aparecerán se dan de forma anónima, con vocación de código secreto. Un pequeño ejiercicio de comunicación cómplice que espero que cada destinatario llegue a recibir. Con un poco de suerte, alguno para el que no era, se dará por aludido y asi me ahorraré tener que dar explicaciones de más. No obstante, habrá alguna honorable excepción a esta norma, y me gustaría empezar por una de las mismas.

Sin duda que esta tesis habría sido imposible sin mi director Humberto Bustince, que primero me mandó a Granada (sin conocerme de nada) y luego me acogió sin pensarlo dos veces. Fue un poco lata porque yo llevaba una entrevista de trabajo preparada al dedillo, pero no tuve que recitar nada de lo que llevaba aprendido. En

segundo lugar, agradezco a mi codirector, Óscar Cordón, por haberme transmitido una extraña (pero muy poco lucrativa) pasión por la investigación en sus clases del máster, y en general, por responder a los correos tan rápido. Creo que la huella de ambos directores se nota durante estas páginas, y sin ellos, ninguna de ellas hubiese sido posible.

Agradezco tambien, sin ningún orden en particular, pues quizá muchos de ellos ni lleguen a leer nunca estas lineas:

- La mejor residente de Zaragoza, por toda una vida (adulta) de confidencias, lloros y buenos momentos.
- Al consejo de sabios.
- La parte más bonita de México, por la bicicleta regalada que siempre conservaré en mi corazón.
- Los dos mosqueteros, por un océano de favores y comprensión.
- A mi compañero de pelis favoritas y a mi médico particular. Los lazos de sangre no valen en nada en comparación con el valor de una vida juntos y un universo propio compartido.
- A mi gallega favorita.
- Al mejor becario que Google ha dado nunca.
- Al mejor informático y compañero que con el que compartir cervezas.
- A la cuadrilla, o la banda, o lo que queramos ser. Por haberme enseñado partes del mundo con vuestro corazón de oro. Nunca volveré a ser el mismo después de conoceros (el cambio es a mejor).
- A la mejor futura vicerrectora. Da igual lo que digas, vas camino de ello.
- En mi mesilla de Colchester siempre había una luz y una foto impresa en una polaroid (bueno, no era una polaroid, era una marca más barata). A toda la gente que aparece en esa foto. Y una más.
- A Danielinho. Por una presencia casi fantasmagórica que hace de su propia existencia un mito.

Y finalmente, como bien sabe todo el mundo, junto con el primer autor de un artículo, hay que mirar el último, que es el que dirige y el que manda. En este caso, no pueden ser otros que mis padres. Nacer en una familia que te quiere, te aprecia, te cuida y te hace llegar llegar a ser tu mejor yo, es el mejor regalo que Dios te puede dar. Yo tengo la inmensa fortuna de haberlo recibido, y espero algún día de estar a la altura de tamaña oportunidad. Y con permiso de mamá, voy a agradecer especialmente a papá mis primeros años en Pamplona, por los innumerables viajes a la estación de buses de Zaragoza. Nunca podré repagar un gesto así.

Y para despedirme, me gustaría acordarme de mi abuelo Luis Idocin. Nunca llegaste a estar, y de alguna manera, siempre estás ahí.

Gracias por todo. A todos.

Contents

Part I

Thesis

1

Introduction

Je ne crains pas les hommes d'armes ;
ma voie a été tracée devant moi. S'il y a
des hommes d'armes, Monseigneur Dieu
m'ouvrira une voie pour aller vers
Monseigneur Dauphin. Pour cela je suis
venu.

Sainte Jeanne d'Arc

INformation fusion involves the combination of multiple sources of information to obtain a more accurate and thoughtful knowledge of a phenomenon. This is one of the key components in many fields where large volumes of data are digitised, as the expressivity of a suitable fusion process can significantly surpass the information present in any of the individual sources (1, 2). One of the difficulties of the information process is that the data to integrate might come from heterogeneous sources or can vary significantly in its nature. For example, an image is captured using many sensors of the same kind, but a car makes decision-making using different sensors, such as air pressure or video. Contrastive learning and stable diffusion models integrate both image and text in their training process (3).

As the nature of information can vary significantly, different structures and models should be built according to the original nature of the data to properly exploit its features. One of the theoretical frameworks developed to address this problem is aggregation theory (4). An aggregation function $A : [0,1]^n \to [0,1]$ is defined as such if the following conditions hold:

- A(0, . . . , 0) = 0
- A(1, . . . , 1) = 1

- A is increasing.

There have been a significant number of monographics on this topic covering some of the most important aggregation function families and some extensions of this concept to cover weaker monotonicities or numeric domains (5, 6).

The way in which information is stored can also open new opportunities to capture and represent its particularities. Non-structured data, like audio, video or text has been profoundly studied in the field of neural networks, specially in deep learning models (7, 8). These networks transform the original data into a a numerical vector of real values from which classical machine learning methods, can be applied (9). Although extremely popular, they are not the only methods to extract and transform such information into a vectorial representation. For example, Common Spatial Patterns (CSP) and related methods are also very popular to do so in signal processing (10), especially in Brain Computer Interfaces (11, 12).

Social networks have also been used to represent data that has strong codependencies, interactions and causality relations (13). For example, network science has been used in marketing and user behaviour modelling in social networks, where people are naturally connected to each other (14, 15). It has also been used to study the functional and structural connectivity of multi component systems, such as the brain (16). The amount of information that this network contains depends on the structure, the edge weights, and possibly the node attributes. In the latter case, not only should the methods take into account the structural properties of the nodes but also the attributes contained in each actor (17, 18). This is a particularly important consideration for the case of task of node embedding, in which each node of the network is represented as a numerical vector (19). In this case, the algorithm that performs the embedding must take into account both similarities in the original attributes and the structural properties of each one. This means that the information reduction is also bigger, as two different information must be joint together in the same representation.

There are other cases where information fusion is performed in a more subtle way. For example, it is possible to explore different strategies in the training process of a neural network in order to incorporate additional information to the main task. For example, additional embeddings and an additional loss function can be used in a standard training procedure for a classification problem. This strategy uses more information for each sample us, which makes the training more efficient in terms of epochs (20, 21). Other examples of this are reinforcement or distillation learning, which consist of fusing the preexisting information of one model with the new data generated in the training procedure (22, 23).

As general as the fusion methods existing can be, there are still significant gaps in the existing literature to be covered, and some questions whose answers are not completely satisfactory. The search for the optimal aggregation function has been studied before using penalty functions and moderate deviations, which measure the similarity of all possible inputs with regards to penalty or a similarity measure (24, 25). However, they still require fine tuning some parameters and they do not necessarily find the optimal results in a real world problem. Besides, the kind of data used to solve the problem can deeply affect the final performance of the system (26). Using numbers, intervals, fuzzy sets or any of its generalisations might also carry additional complexities in their fusion procedures. Some of these complexities have not been yet studied, and not all of the operators used in specific numeric domains have been extended to other types of data, like intervals or real numbers.

The possibilities of information fusion in more complex structures, like social networks, offers more possibilities and gaps to research. Homophily, the tendency of an actor to relate to similar actors, has been studied in depth (27, 28). However, there are more structural traits that could be exploited in different applications. The relationship between the attributed nodes and its structural connections is also unexplored in many works, where only the distance from the cosine is considered to compare different actors (29, 30).

The aim of this dissertation is to provide further results to elaborate more refined answers the question of how to find the best fusion processes in a system, and how it affects the rest of its components. In order to do so, we have studied how a wide range of aggregation functions behave in different classification problems, using both standard tabular data, time series and deep learning features. We have also develop a new extension of penalty-based aggregation functions and moderate deviation functions for interval-valued data, designed to tackle some of the problems or aspects that were present in such classification systems. We have used these concepts in Motor-Imagery BCI systems, which consist of series of signal processing steps and a final decision making scheme (31, 32).

Regarding social network analysis, we have a developed a new kind of functions, the affinity functions, which exploit different social properties of the network to represent edge weights. We have also proposed a series of algorithms to exploit their properties to perform community detection, detect important nodes, or to exploit external information of the network. Finally, these methods were used in the area of digital humanities to provide novel results in comparative mythology and artistic image classification.

2

Motivations and objectives

Si no consigo un barco, iré nadando

San Francisco Javier

2.1 Motivation

The high performance of machine learning models for tasks like face recognition, image and video generation, and voice-to-text, relies on the huge volumes of data available in modern day computing (33, 34). State-of-the-art deep learning systems are comprised of millions of parameters, which require even larger amounts of data (35). These data can come from different sources of information, and sometimes models can be trained using different types of data. For example, Autoencoders like CLIP (3) or generative models like DALL-E use both images and text to train (36). Another paradigm rises when many models are used to solve one problem, in which case the final decision must be performed taking into account their individual traits and the results of their interaction (37, 38).

Huge efforts have been devoted to research these topics, however, there are still some important gaps between theory and practice that need to be addressed. Practical research has focused on the technical possibilities and techniques that allow a computer or a distributed system to deploy models with millions of parameters and feed them with TBs of information (39, 40, 41). There has also been important research to perform knowledge extraction on non-vectorial data representations. Some of the most popular structures of this kind are social networks (42), time series (43) and more complex data flows, like the forward and backward passes of a neural network (44). Deep learning has been very successful to extract such representations from image, text, and even graphs. Graph convolutional neural networks have been also applied in social network representations to perform classification and clustering problems (45).

On the other hand, theoretical research has focused on the functions that can be used to model the uncertainty of the data, or the mathematical properties that induce some desirable traits in the aggregation process (5). These functions lack the power of complex models, but offer behaviour that can be understandable. In a similar fashion, fuzzy logic inference systems have been used successfully in problems where explainability was an important requirement in the problem solution (46). These theoretical approaches, however, has been devoted mostly to study vectorial tabular data.

Regarding the relationship between both approaches, theoretical research on deep learning is still not as developed as its practise. For example, explainability of deep learning models is considered using other experimental methods (44). The practical approaches of theory of fusion data on the other have not escalated to big deep learning models, even though some notable works have been developed (47). They have been, however, very successful in ensemble learning and decision-making (37, 48), including big data (49). Although the reason for the success and failures of some aggregation functions in the practical domain is not yet clear.

Consequently, we think that it is of interest to further expand the possibilities of actual theoretical results in aggregating and combining information, both in the case of vectorial and non-vectorial representation of data. Expanding the actual existing methods regarding aggregation theory can also be useful in order to expand the frontier of successful tasks handled using this line of work. Finally, as the systems used to solve real-life problems are composed of many components, the study of some of these functions in modelling such interactions can also fuel further investigation.

2.2 Objectives

The main objective of this dissertation is: *To develop new ways to extract and fuse features from multicomponent systems, where the sources of available information are heterogeneus.*

For its achievement, we set the following particular objectives:

- To study the behaviour of different aggregation functions under such conditions and propose new ones according to the needs for the frameworks where they are used.

- To define the appropriate set of functions to characterise the different information encoded in the edges of a social network.
- To study these function also in interval-valued data.
- To apply such functions to real problems: BCI-MI, tabular classification, social network analysis and artistic image classification.

3

Discussion of research findings

Les enfants disent que les gens sont
parfois pendus pour avoir dit la vérité.
Sainte Jeanne d'Arc

Hable poco, y en cosas que no es
preguntado no se meta.
San Juan de la Cruz

IN this section we shall recall the most important notions in each of the results obtained in this PhD thesis. Each of them has been published, accepted or at is currently under review in a scientific journal. The following subsections break up those results into the papers where they were reported.

3.1 Motor-Imagery-Based Brain–Computer Interface Using Signal Derivation and Aggregation Functions

In this work we present the Enhanced Multimodal Fusion Framework (EMF), which is a Motor-Imagery Brain-Computer-Interface (MI-BCI) classification framework. The new method proposed elaborates further on a previous work (50), in which the authors proposed a decision making scheme to perform the final decision of the system based on multiple classifiers, each of them trained in different wave bands.

Figure 3.1 shows a scheme of of the EMF. The framework is divided into tree main components:

- Signal processing: this step consists of capturing the signal and computing the Fast Fourier transform (FFt) in order to obtain the desired wave bands. Then, we compute differentiation of the signal. This differentiation of the signal is similar to the Laplacian transforms performed in the literature (51, 52). However, its effects on the signal can be more easily explained if we decompose the signal in the three classical components of time series: trend, stationary and random components, from which the differentiation only preservs the last one.

- Feature extraction: we compute the CSP on the differentiated signals, and then, we use those features to train different classifiers. We train five kinds of classifiers: LDA, QDA, KNN, SVM and GP. For train one of each kind for all the wave bands studied.

- Multimodal decision making: finally, we compute the final decision using two aggregation functions. First, we use an aggregation function to fuse the outputs obtained for each wave band by each kind of classifier. Then, we fuse the outputs of that fusion step using another aggregation function. We obtained the final decision in this step.

The novelties of the EMF with respect to other MI-BCI classification frameworks are the multimodal decision making phase, consisting on two aggregation functions, and the signal differentiation phase. We also used a higher number of CSP features compared to the standard in the literature, which also improved the performance of the EMF (53).

With respect to the multimodal decision, not only we increased the number of aggregation functions to two instead of one, but we tested a wide range of possible aggregation functions to use. We discovered that the best results were obtained using different aggregation functions for the two steps of the process, and that a fuzzy integral followed by a n-ary overlap function performed best. The total amount of aggregation functions tested included: T-norms and T-conorms, N-ary overlaps (54), Choquet integral and their generalizations (55), Sugeno integral (56) and two generalizations and OWA operators (57).

The Signal differentiation phase filters the signal, removing the trend and the season component, inherent to a time series (Figure 3.2). We suspected that these components were not very useful, because the best wave bands to discriminate the task were the highest frequency ones considered. This procedure is also convenient to remove possible long-range artifacts in the measurements of the EEG signal. We also showed that this problem is also present in other kinds of neural activation measurements (58).

Framework	Accuracy
EMF	83.15%
KLRRM + LSVM (60)	74,43%
CSP/AM-BA-SVM (61)	78.55%
Dempster-Shafter (62)	81.93%

Table 3.1: Comparison of different BCI frameworks in the four classes task.

The EMF was tested on the public dataset 2a of the BCI Competition IV (59), obtaining very favourable results compared to other alternatives (Table 3.1).

As final conclusions of this work, we state that:

- A flexible aggregation process can improve significantly the performance of the system in some cases where signal processing could not.

- The success of existing signal filtering methods in the BCI literature can be explained in terms of temporal series. The signal differentiation here used is simpler and as good performant as other more complicated processes used due to the same properties that can explain their good performance.

The section of this thesis is associated with the following publication:

Fumanal-Idocin, J., Wang, Y. K., Lin, C. T., Fernández, J., Sanz, J. A., & Bustince, H. (2021). Motor-imagery-based brain-computer interface using signal derivation and aggregation functions. *IEEE Transactions on Cybernetics*.

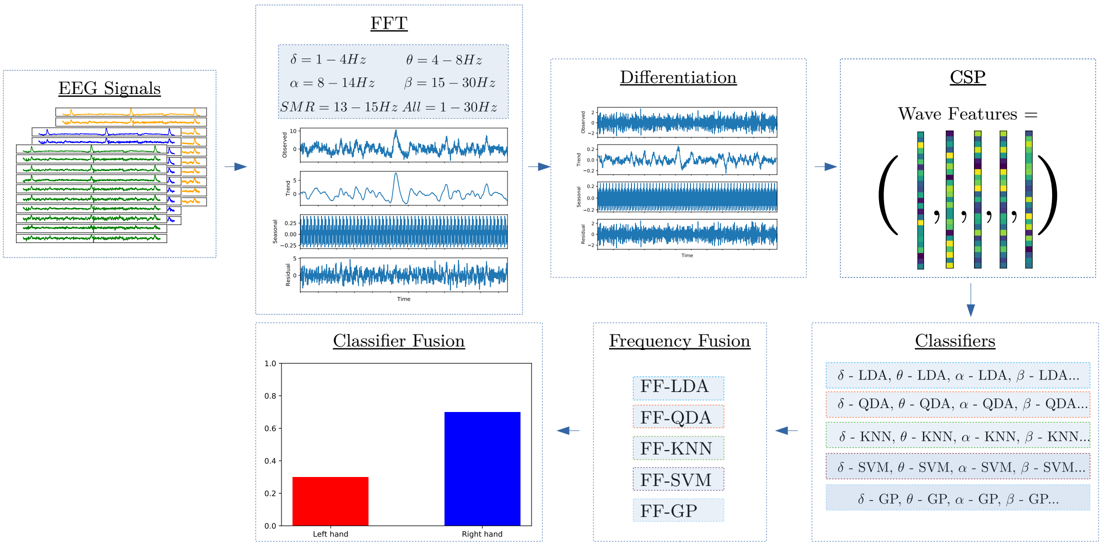
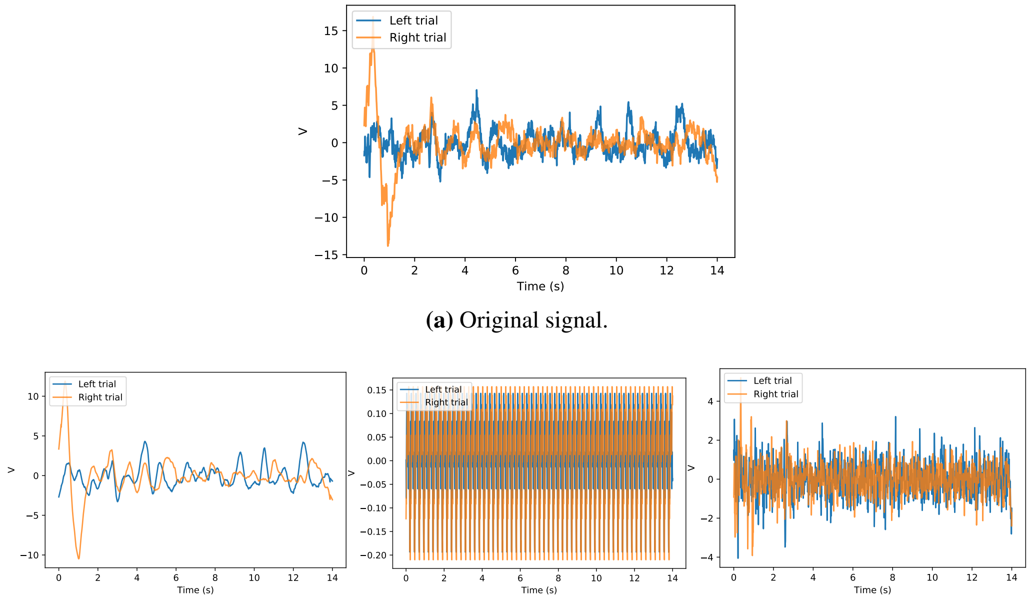

EEG Signals
FFT
$\delta = 1 - 4Hz$ $\theta = 4 - 8Hz$
$\alpha = 8 - 14Hz$ $\beta = 15 - 30Hz$
$SMR = 13 - 15Hz$ $All = 1 - 30Hz$
Observed
Trend
Seasonal
Residual
Time
Differentiation
CSP
Wave Features =
Classifiers
δ - LDA, θ - LDA, α - LDA, β - LDA...
δ - QDA, θ - QDA, α - QDA, β - QDA...
δ - KNN, θ - KNN, α - KNN, β - KNN...
δ - SVM, θ - SVM, α - SVM, β - SVM...
δ - GP, θ - GP, α - GP, β - GP...
Frequency Fusion
FF-LDA
FF-QDA
FF-KNN
FF-SVM
FF-GP
Classifier Fusion
1.0
0.8
0.6
0.4
0.2
0.0
Left hand
Right hand

Figure 3.1: Architecture of the proposed Enhanced Fusion Framework.



(a) Original signal.

(b) Trend.

(c) Stational comp.

(d) Random comp.

Figure 3.2: Signal comparison using the average over all the waves for one subject in the UTS dataset. Original time series and its decomposed components.

3.2 Interval-Valued Aggregation Functions Based on Moderate Deviations Applied to Motor-Imagery-Based Brain Computer Interface

In this work we proposed to extend the concept of Moderate Deviation to interval-valued data. Intervals are a basic form to represent uncertainty, in which we use its width to measure how reliable is an observation. Then, we will exploit this property in a BCI-MI framework.

The notion of Moderate Deviation had been previously developed as a method to aggregate a vector of values, denoting as output of this aggregation the "most similar" point to all of the inputs (24). This concept of "most similar" is expressed using a moderate deviation function:

Definition 1 *A function* $\mathcal{D} : [0,1]^2 \to \mathbb{R}$ *is called a moderate deviation function, if, for all* $x, y \in [0,1]$*, it satisfies:*

(MD1) $\mathcal{D}$ *is non-decreasing in the second component;*

(MD2) $\mathcal{D}$ *is non-increasing in the first component;*

(MD3) $\mathcal{D}(x,y) = 0$ *if and only if* $x = y$*.*

In this work, we expand this notion to an interval-valued setting in this way:

Definition 2 *Let* $\leq_L$ *be a total order on* $L([0,1])$ *and*

$$L(\mathbb{R}) = \{A = [\underline{A}, \overline{A}] \mid \underline{A}, \overline{A} \in \mathbb{R}, \underline{A} \leq \overline{A}\} \tag{3.1}$$

A function $D : (L([0,1]))^2 \to L(\mathbb{R})$ *is called an interval-valued moderate deviation function w.r.t.* $\leq_L$*, if, for* $X, Y \in L([0,1])$*, it satisfies:*

(MD1) D *is non-decreasing in the second component w.r.t.* $\leq_L$*;*

(MD2) D *is non-increasing in the first component w.r.t.* $\leq_L$*;*

(MD3) $D(X,Y) = 0_L$ *if and only if* $X = Y$*.*

Once the concept of interval-valued moderate deviation function has been established, we also show how can this be applied to construct an interval-valued moderate deviation-based aggregation. The idea is the same as with numerical data: given a vector of intervals we choose as the output of the aggregation the interval that is "most similar" to the inputs.

In this work we also give some construction methods in order to apply these concepts in a practical setting. In the following, we display the theorems needed to give a construction method of w-preserving interval-valued moderate deviation functions.

In (24) (Theorem 6) introduced a construction of a moderate deviation function $\mathcal{D} : [0,1]^2 \to [-M_n, M_p]$ in the following way:

$$\mathcal{D}(x,y) = \begin{cases} M_p - M_p R_1(x,y), & \text{if } x \leq y, \\ M_n R_2(x,y) - M_n, & \text{if } x > y, \end{cases} \tag{3.2}$$

for all $x, y \in [0,1]$, where $M_n, M_p \in]0,\infty[$.

Theorem 1 *Let $\alpha \in [0,1]$, $n \in \mathbb{N}$, let M_p, M_n be positive real numbers and $\mathcal{D} : [0,1]^2 \to [-M_n, M_p]$ be a moderate deviation function. Let $F : L([0,1])^{n+1} \to \mathbb{R}$ be the function given, for all $X_1,\ldots,X_n,Y \in L([0,1])$ such that $w(Y) = \min(w(X_1),\ldots,w(X_n))$, by:*

$$F(X_1,\ldots,X_n,Y) = \mathcal{D}\Big(K_\alpha(X_1), K_\alpha(Y)\Big) + \ldots + \mathcal{D}\Big(K_\alpha(X_n), K_\alpha(Y)\Big). \tag{3.3}$$

Theorem 2 *Let $\alpha, \beta \in [0,1]$ with $\beta \neq \alpha$, $\mathcal{D} : [0,1]^2 \to \mathbb{R}$ be a strictly monotone moderate deviation function and $C : [0,1]^2 \to [0,1]$ be an idempotent function non-decreasing in the second component and non-increasing in the first component. Then the function $D : (L([0,1]))^2 \to L(\mathbb{R})$ given by:*

$$\begin{cases} K_\alpha(D(X,Y)) = \mathcal{D}\left(K_\alpha(X), K_\alpha(Y)\right), \\ w(D(X,Y)) = C\left(w(X), w(Y)\right) \end{cases}$$

is a w-preserving interval-valued moderate deviation function w.r.t. $\leq_{\alpha,\beta}$.

The following corollary gives us a method of constructing wD-means based on Theorem 2 and Theorem 1.

Corollary 1 *Under the assumptions of Theorem 1, where $\mathcal{D} : [0,1]^2 \to \mathbb{R}$ is given by Equation (3.2) with R_1, R_2 being continuous strictly monotone restricted equivalence functions and $D : (L([0,1]))^2 \to L(\mathbb{R})$ is given by Theorem 2, the following statements are equivalent:*

(i) $F(X_1,\ldots,X_n,Y) = 0$;

(ii) $M_D(X_1,\ldots,X_n) = Y$, *where the interval-valued wD-mean M_D is given by Equation* (**??**) *with* $B = \min$;

(iii)

$$\sum_{i=1}^{k} \left(M_p - M_p R_1\Big(K_\alpha\left(X_{\sigma(i)}\right), K_\alpha\left(Y\right)\Big)\right) + \sum_{i=k+1}^{n} \left(M_n R_2\Big(K_\alpha\left(X_{\sigma(i)}\right), K_\alpha\left(Y\right)\Big) - M_n\right) = 0 \tag{3.4}$$

where $\sigma : \{1,\ldots,n\} \to \{1,\ldots,n\}$ *is a permutation such that* $X_{\sigma(1)} \leq_{\alpha,\beta} \cdots \leq_{\alpha,\beta} X_{\sigma(n)}$ *and* k *is the greatest number from* $\{1,\ldots,n\}$ *satisfying*

$$\sum_{i=1}^{n} \mathcal{D}(K_\alpha(X_{\sigma(i)}), K_\alpha(X_{\sigma(k)})) \leq 0. \tag{3.5}$$

We used the interval-valued moderate deviation-based aggregation in a MI-BCI framework in order to test its suitability to aggregate data.

First, we construct the intervals from the probability for each class obtained from each classifier. We use the length of the intervals to measure the inaccuracies or uncertainties related to these classifiers' outputs. To do so, we have used the mapping given in (63):

$$F(x,y) = [\underline{I(x,y)}, \overline{I(x,y)+y}] \tag{3.6}$$

where I is an fuzzy implication function, x is the probability for each class obtained from the classifier and we set y as 0.3. We crop the values so that they are contained in the $[0,1]$ interval. We have tried three different fuzzy implication functions to construct the intervals:

- Łukasiewicz: : $I(x,y) = \max(1-x,y)$.
- Reichenbach: $I(x,y) = \min(1, 1-x+y)$.
- Kleene-Dienes: $I(x,y) = 1-x+xy$.

We tested the behaviour of our new approaches in the BCI competition IV dataset 2a (IV-2a) and the BCI competition IV dataset 2b (IV-2b), which are detailed in (64). We compared this against interval-valued OWA operators and other numerical aggregation functions, obtaining favourable results to our interval-valued MD aggregation (Table 3.2) using a traditional fusion scheme and a hierarchical one (50).

Proofs related and more information about this work can be found in the related publication of this results:

Fumanal-Idocin, J., Takáč, Z., Fernández, J., Sanz, J. A., Goyena, H., Lin, C. T., Wang, Y.K., & Bustince, H. (2021). Interval-Valued Aggregation Functions Based on Moderate Deviations Applied to Motor-Imagery-Based Brain–Computer Interface. *IEEE Transactions on Fuzzy Systems*, 30(7), 2706-2720.

Table 3.2: Performance for the Left/Right classes task using different IV-aggregations and BCI frameworks.

Dataset	Classifier	Accuracy
IV-2a	Trad. Average	0.8119 ± 0.0384
	MFF Choquet	0.8049 ± 0.0417
	MD_2 (Trad.)	0.8251 ± 0.0538
IV-2b	Trad. Average	0.7213 ± 0.0824
	MFF Choquet	0.7440 ± 0.0058
	MD_2 (Trad.)	0.7366 ± 0.0708

3.3 Multi-cost penalty functions applied to Motor-Imagery based Brain-Computer-Interface

In this work we presented a generalisation of penalty-based aggregations applied to a new BCI framework, that we called Multi-Cost Aggregation choosing functions (MCA). Similarly to penalty-based aggregations, MCAs use a cost function to select the best aggregation function for a given set of inputs. One of the key differences between them is that MCA use a more general function than a penalty function to choose the optimal aggregation function.

In order to construct a MCA, we start by introducing the notion of Quasi-Restricted Similarity and Dissimilarity functions. These functions are based on the concepts of REF and RDF given in (65).

Definition 3 *Let $n \geq 1$. A Q-REF function is a function $H_s : [0,1]^{n+1} \to [0,1]$ such that:*

$$H_s(X,y) = H_s(x_1,\dots,x_n,y) = 1 \text{ if } x_1 = \cdots = x_n = y. \tag{3.7}$$

Note that REFs are specific instances of Q-REF functions. And analogously:

Definition 4 *Let $n \geq 1$. A Q-RDF function is a function $H_d : [0,1]^{n+1} \to [0,1]$ such that:*

$$H_d(X,y) = H_d(x_1,\dots,x_n,y) = 0 \text{ if } x_1 = \cdots = x_n = y. \tag{3.8}$$

We also gave a general method to build Q-REF and Q-REF functions as follows.

Proposition 1 *Let $h_{s1},\dots,h_{sn} : [0,1]^2 \to [0,1]$ be a family of Q-REF functions and let $A : [0,1]^n \to [0,1]$ be an aggregation function. Then, $H_s^A(x_1,\dots,x_n,y) = A(h_{s1}(x_1,y),\dots,h_{sn}(x_n,y))$ is also a Q-REF function.*

Proposition 2 *Let $h_{d1},\dots,h_{dn} : [0,1]^2 \to [0,1]$ be a family of Q-RDF functions and let $A : [0,1]^n \to [0,1]$ be an aggregation function. Then, $H_d^A(X,y) = A(h_{d1}(x_1,y),\dots,h_{dn}(x_n,y))$ is also a Q-RDF function.*

One of the main ideas of this work, is that instead of using only one function, we combine two functions to create a new cost function that takes into account both criteria. We can do so by combining Q-REF and Q-RDF in the following way:

Proposition 3 *Let $H_{s1},H_{s2} : [0,1]^n \to [0,1]$ be two Q-REF functions. Then, for every $\alpha \in [0,1]$*

$$\alpha H_{s1} + (1-\alpha)H_{s2} \tag{3.9}$$

is also a Q-REF function.

Proposition 4 *Let $H_{d1},H_{d2} : [0,1]^n \to [0,1]$ be two Q-RDF functions. Then, for every $\alpha \in [0,1]$*

$$\alpha H_{d1} + (1-\alpha)H_{d2} \tag{3.10}$$

is also a Q-RDF function.

We can also consider the convex combination of a Q-REF and a Q-RDF function. If $x_1 = \cdots = x_n = y$, we have that:

$$\alpha H_d(x_1,\dots,x_n,y) + (1-\alpha)H_s(x_1,\dots,x_n,y) = 1-\alpha \tag{3.11}$$

So:

Proposition 5 *Let $H_d,H_s : [0,1]^n \to [0,1]$ be a Q-REF and a Q-RDF function, respectively. Then, for any $\alpha \in [0,1[$, the function:*

$$H(X,y) = \min(\frac{\alpha H_d(X,y) + (1-\alpha)H_s(X,y)}{1-\alpha},1) \tag{3.12}$$

is a Q-REF function.

We have considered a set of Q-RDFs and Q-REF measures as cost functions. Depending on the mixed functions, their convex combination is also a Q-REF or a Q-RDF. Given a vector X of size n, where each element of X is contained in the unit interval the Q-RDFs measures studied are the following:

- Huber loss:

$$h(x_i, y) = \begin{cases} (x_i - y)^2 & (x_i - y)^2 \leq M \\ 2 * M * (x_i - y)^2 - M * M & (x_i - y)^2 \geq M \end{cases} \tag{3.13}$$

  where $H(X,y) = \frac{1}{n}\sum_{i=1}^{n} h(x_i, y)$.

  (We use $M = 0.3$ for our experimentation)

- Quadratic cost:

$$H(X,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i - y)^2 \tag{3.14}$$

- Optimistic cost:

$$H(X,y) = (max(X) - y)^2 \tag{3.15}$$

- Pessimistic cost:

$$H(X,y) = (min(X) - y)^2 \tag{3.16}$$

The Q-REF measure studied is:

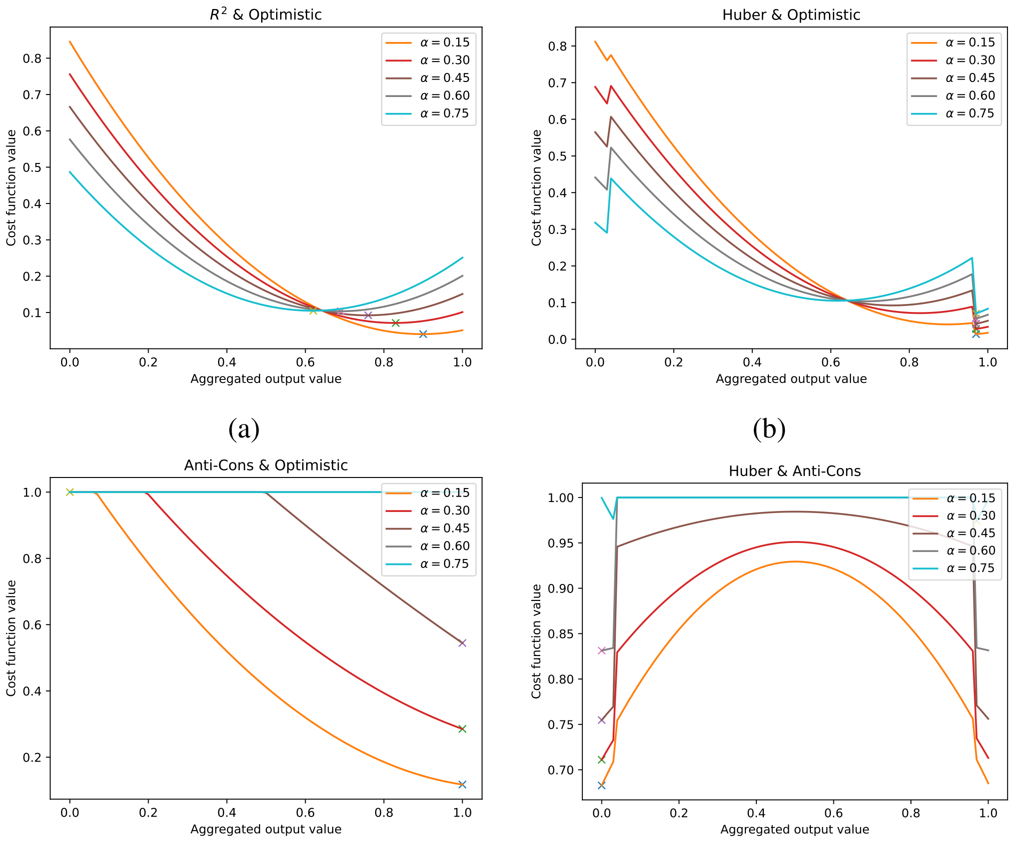
- Anti-consensus cost:

$$H(X,y) = \frac{1}{n}\sum_{i=1}^{n}(1 - (x_i - y)^2) \tag{3.17}$$

Combining different Q-REF and Q-RDF results in different behaviours, as illustrated in Figure 3.3. A visual scheme of the aggregation process using a MCA function, can be used in Figure 3.4.

The BCI framework used in this paper consists of extracting features from 4 different wave bands from EEg signals: $6-10$, $8-15$, $14-28$ and $24-35$ Hz. Then, we apply Spectral Spatial Decomposition and Common Spatial Patterns to extract features from the wave bands. We use LDA classifiers to train on those features obtain an output for each wave band. Then, we fuse the output of these classifiers using a MCA function. This framework is illustrated in Figure 3.5. We also applied this fusion procedure with a multiscale CSP feature extraction (66).

Table 3.3 shows the results for the the CBCIC 2020 dataset (67). This dataset consists of brain imaging signals from 10 hemiparetic stroke patients with hand functional disability in a rehabilitation task. The data contains 80 diferent trials of left/right hand movements. Decoding motor cortical signals of brain-injured presents several challenges as the presence of irregular because of the altered neurodynamics.



(a)

(b)

(c)

(d)

Figure 3.3: Effect of different α parameters for a vector of five, randomly chosen numbers $\in [0,1]$: $(0.60, 0.85, 0.61, 0.52, 0.52)$ when using different combinations of functions as cost functions. The $\times$ marks the minimum for each α parameter in each error configuration.

Table 3.3: Accuracy results for the proposed framework in the CBCIC 2020 dataset.

Dataset		Quadratic	Optimistic	Huber	Pessimistic
CBCIC	Optimistic	0.8123			
	Huber	0.8215	0.8142		
	Pessimistic	0.8113	0.8000	0.8224	
	Anti-consensus	0.8215	0.8221	**0.8231**	0.8215

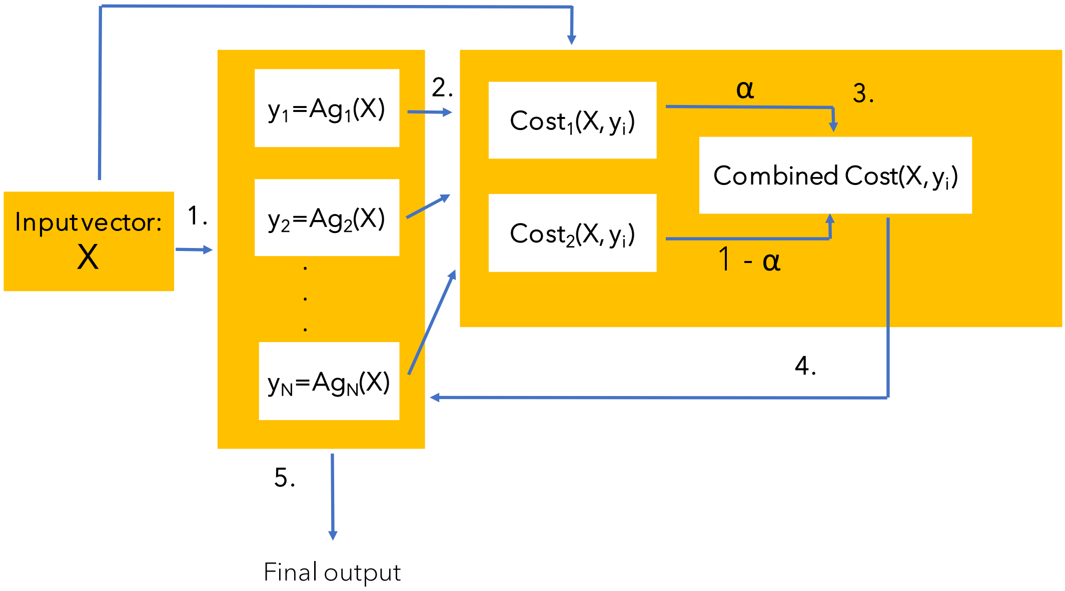


Figure 3.4: Visual scheme for the MCA aggregation process. (In the case of the BCI framework in Fig. 3.5, X is the output of the LDA classifiers). **1**. We compute all the possible aggregations. **2**. We compute both cost functions for each aggregation output (y_i) with respect to the input vector. **3**. We combine both costs for each aggregation with the mixing parameter α. **4**. We select the aggregation with the least cost value. **5**. That aggregation is the final output of the MCA.

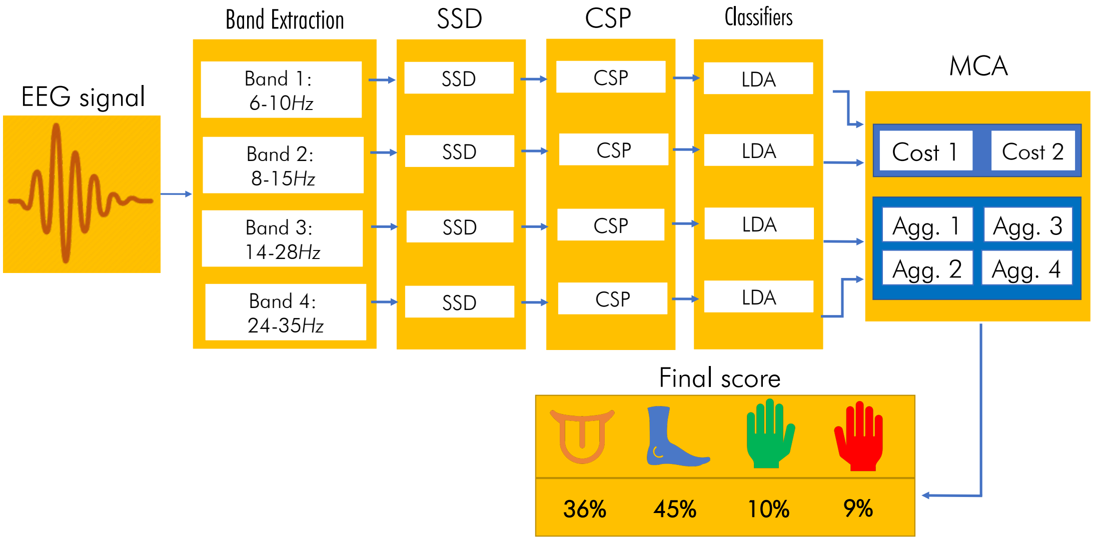


Figure 3.5: Visual representation of the framework used in this study. First, we measure the EEG band, and extract the information from four different frequency bands. Then, we apply SSD and subsequently CSP to reduce dimensionality and extract features from each band. From each frequency band we train a different LDA classifier. We make a final decision by aggregating the output from all the LDA classifiers using a MCA, which results in the estimated probabilities for each one of the possible classes.

3. DISCUSSION OF RESEARCH FINDINGS

More details can be found in the following paper: Fumanal-Idocin J., Vidaurre C., Fernandez J., Gomez M., Andreu-Perez J., Prasad M. & Bustince H. (2023) Supervised Penalty-based Aggregation Applied to Motor-Imagery based Brain-Computer-Interface *Pattern Recognition*

3.4 A Generalization of the Aggregation of Forces in the Gravitational Clustering to Perform Anomaly Detection

In this work we propose a generalisation of the gravitational clustering algorithm (68) using the novel concept of almost aggregation function, which is an extension of the concept of aggregation function to real positive numbers.

The gravitational clustering algorithm (68) employs the Newton gravitational law within the process of clustering. The scheme of this original algorithm is as follows.
Assume that we have N particles $p_1,\dots,p_N$, with their positions $s_1,\dots,s_N \in \mathbb{R}^n$.

1. Initially we:

   - assign a mass 1/N to each particle p_i,
   - fix two real positive parameters ε and δ. We utilize δ for determining the actual time step longitude dt. Specifically, dt has to be such that during the time slot $[t,t+dt]$ the fastest particle displacement is equal to δ. Besides, we use ε to determine how close two particles can be before they are merged into one single particle, see step 2 (ii) below,
   - If in a moment two particles find themselves in a distance less than ε, we unify them into one single particle, with the mass equal to sum of masses of both of them and position done by their center of gravity.
   - Set initial time $t = 0$.

2. We repeat the following steps (i)-(iv) until one single particle remains.

   (i) In each time interval $[t,t+dt]$, for each particle i we compute its movement influencing function:

$$g(i,t,dt) = \frac{1}{2} G \sum_{j \neq i} \frac{m_i(t) m_j(t)}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|^3} dt^2 \tag{3.18}$$

   where G is a positive constant.

   (ii) For each particle i, its new position is:

$$s_i(t+dt) = s_i(t) + g(i,t,dt) \tag{3.19}$$

   (iii) We raise t to $t+dt$.

(iv) If two particles i and j are in a distance less than ε, we unify them into one single particle, with its mass equal to sum of masses of both of them and its position given by their centre of gravity of the two original particles.

Finally, we have just one particle. The relative life of the configuration with k clusters can be computed as

$$R_k = \frac{t_{k+1} - t_k}{T} \tag{3.20}$$

We choose as solution that corresponds to the configuration with largest relative life.

This model described above can be generalized by using a more general expression for particle movement governing function instead of (3.18). This was already done in (68) by using the following family of expressions

$$g(i,t,dt) = \frac{1}{2} G \sum_{j \neq i} \frac{m_i(t)^a m_j(t)^b}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|^3} dt^2 \tag{3.21}$$

with $a, b > 0$.

In particular, it was shown that from an experimental point of view, the best results are obtained for $a = b = 0$. The resulting gravitational model is called Markov unitary model and it is described by the governing function:

$$g(i,t,dt) = \frac{1}{2} G \sum_{j \neq i} \frac{1}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|^3} dt^2 \tag{3.22}$$

We proceed now to define an almost aggregation function:

Definition 5 *An n-dimensional function* $M : ([0,\infty[)^n \to [0,\infty[$ *for* $n \geq 2$ *will be called almost aggregation function on the interval* $[0,\infty[$ *if:*

1. $M(0,\ldots,0) = 0$.

2. M *is increasing in each variable (usual monotonicity).*

Note that every existing aggregation function is also an almost aggregation function.

We use this kind of functions to substitute the Summatory in Eq. 3.22 for a more general expression:

$$\mathbf{g}(i,t,dt) = \sum_{j \neq i} \frac{m_i(t) m_j(t)}{m_i(t)} \frac{\mathbf{s}_j(t) - \mathbf{s}_i(t)}{|\mathbf{s}_j(t) - \mathbf{s}_i(t)|^3} dt^2 \tag{3.23}$$

Observe that the product of the masses in the gravitational algorithm always involves two integer masses. In order to generalize this discrete product, given an integer $N \geq 1$, we are going to take a function: $H\{1,...,N\} \times \{1,...,N\} \to \mathcal{H} \subset \mathcal{R}$, such that:

- H is increasing.

- H is symmetric.

- H is bounded on $\{1,...,N\} \times \{1,...,N\}$.

As H functions, we used the following family of functions:

$$H_c(x,y) = \begin{cases} b_1 & \text{if } (xy)^c < b_1, \\ b_2 & \text{if } (xy)^c > b_2, \\ (xy)^c & \text{if } B_1 \leq (xy)^c \leq b_2 \end{cases} \tag{3.24}$$

These functions can be adjusted to each situation by changing the b_1, b_2, which can adapt the function according to the behaviour of the algorithm. We use this property to speed up the simulation if particles are too far from each other, and to slow it down if they are very close to each other.

So, the final function computed instead of 3.18s is the following:

$$\mathbf{g}(i,t,dt) = A_{j \neq i}(\frac{H(m_i(t),m_j(t))}{m_i(t)} \frac{\mathbf{s}_j(t) - \mathbf{s}_i(t)}{|\mathbf{s}_j(t) - \mathbf{s}_i(t)|^3})dt^2 \tag{3.25}$$

This expression is considerably more general than the original expression used in (68). One of the most interesting properties We exploit this in order to adapt this algorithm to perform anomaly detection instead of clustering.

In order to do so we are based on the idea that in the process of the gravitational simulation, those particles that spent a lot of time of the simulation isolated are likely to be anomalies. Besides, those particles, if they join others in clusters, tend to be clusters with few samples. Based on those two conditions, we can detect those observations in the simulation that followed the expected behaviour of an anomaly.

When applying the model to a new observation, we use a 1-NN classifier, so that the new particle will be classified depending on its closer particle using the Euclidean distance.

We have compared the results obtained with our newly proposed gravitational algorithm with three other classical anomaly detection algorithms using 4 datasets taken from Keel (69):

1. Local Outlier Factor (70) estimates the data density near one point, based on its k nearest neighbours. Points that have substantially lower density than its neighbours are considered as anomalies.

Table 3.4: Comparison for different anomaly detection algorithms in five different real-world datasets, using the F1-score as performance metric.

	Ecoli1	Ecoli3	Glass	Wisconsin	Yeast
LOF	0.84029	0.9085	0.8490	0.6881	0.9170
OCSVM	0.6237	0.6303	0.6957	0.8584	0.6676
Isolation Forest	0.8206	0.8748	0.9274	**0.9631**	0.9147
Anomaly-Grav	**0.8634**	**0.9466**	**0.9377**	0.7804	**0.9423**

2. One-Class SVM (71) learns a decision function for anomaly detection, in order to maximize the likelihood of the observed data with respect to the rest.

3. Isolation Forest (72) constructs an ensemble of trees from different subsamples of the original dataset, and then, evaluates the number of partitions for each sample required to isolate the sample in the set of trees.

The publication associated with these results is the following one: Fumanal-Idocin, J., Rodriguez-Martinez, I., Indurain, A., Minárová, M., & Bustince, H. (2022). A Generalization of the Aggregation of Forces in the Gravitational Clustering to Perform Anomaly Detection. *Information Sciences*.

3.5 Community Detection and Social Network Analysis based on the Italian Wars of the 15th Century

Social network dynamics are present in everyday life and in some well known historical events. In this work we study how similar is our modelling of social networks compared to real life, and if real life phenomena can be adapted to solve computational problems.

In order tackle these issues we star by defining the notion of affinity functions as a set of functions in social network analysis of two different actors, Actor$_x$ and Actor$_y$, establishing their mutual relation using C:

$$A_C : [Actor_x, Actor_y] \rightarrow [0,1]$$

Usually, this C is the adjacency matrix that quantifies the relationships in each pair of actors. The affinity between two actors shows how strongly these two are connected. Since affinities are not necessarily symmetrical, the strength of this interaction depends on who the sender and receiver are, as happens in human interaction e.g. unrequited love.

This definition of affinity function has two benefits compared to the standard adjacency matrix:

- Explicit asymmetries in each relationship.
- Universal comparability among edges, since all values are in the same scale.
- Information is locally scaled. So we can interpret one actor relationships without accessing the rest of the network.

We proceed to list some affinity functions:

- **Best Friend affinity**: the affinity of the actor $Actor_x$ over the $Actor_y$ is defined as the percentage of the total connectivity of Y that corresponds to $C_{x,y}$.
- **Best Common Friend affinity**: the affinity between two actors is defined as the biggest affinity common to the both of them. It can be computed using both the adjacency matrix or another previously calculated affinity.
- **Friends Forever affinity**: the affinity of two actors reflects the durability of the relation in time.

Table 3.5: Formula proposed for each of the affinities. C is the adjacency matrix and N is the total number of actors in C.

Best friend	$A_C(x,y) = \frac{C_{x,y}}{\sum_{a=1}^{N} C_{x,a}}$
Best Common friend	$A_C(x,y) = Max\{Min(C_{x,z}, C_{y,z})\}/\sum_{a=1}^{N} C_{x,a}$
Friends forever	$A_C(x,y) = \sum(\frac{C_{x,y}(t)}{\sum_{a=1}^{N} C_{x,a}(t)})\frac{1}{\vert T\vert}, \forall t \in T$
Machiavelli	$A_C(x,y) = 1 - \frac{abs(I_x - I_y)}{Max(I_x,I_y)}, I_a = Sum(Degree(x'))\forall x$ such that $C(a,x') > 0$

- **Social Networking affinity**: the affinity between two actors, Actor$_x$ and Actor$_y$, is based on the affinities of the actors connected to Actor$_x$ with respect to Actor$_y$.

- **Machiavelli Affinity**: the affinity between two actors is based on the social structure that is built around the two of them.

Their correspondent formulas can be found in Table 3.5.

Using the affinity functions, we have developed a community detection algorithm, the Borgia Clustering, based on an important chapter in the European history: the 15th-century Italian Wars. In 1497, Cesare Borgia, under the command of his father, Alexander VI, and as Commander in Chief of the Papal Army, marched through the centre of Italy, conquering all the territories that have been traditionally linked to the Papal States (73). This thrilling moment in Renaissance history provided us with not only memorable moments of unparalleled initiative and wit, but also with an excellent example regarding human interaction in both personal and communitarian levels.

This algorithm is based on the classical Gravitational Clustering algorithm (68). Each actor starts as a different community that gets closer to the others due to the effect of an attraction force. Our contention is that the Borgia Clustering force and particle movement generates communities emulating the dynamics seen during the XV Italian wars.

To represent each actor, we use a combination of the best friend affinity matrix and the best common friend affinity matrix, and an influence matrix, S, based on the former. By using the best friend affinity, we favour strong pair-wise interactions, and with the best common friend affinity, we also favour the formation of communities whose members share a high number of friends. Also, each actor has a “social value”, equal to its number of connections, that reflects its popularity.

The actors attract to each other in a simulation of gravitational-like force, depending on their social value, affinity and distance. We consider that two actors a,b collide when the value of $S(a,b)$ is bigger than $S(b,b)$. Then, we interpret that the actor a has as much influence over the actor b as b has over itself. In that moment they are fused to form a new community. As a result of this, the most affine pairs of actors will naturally join first and start forming communities.

We have decided to construct the Borgia Clustering algorithm by modifying the classical gravitational algorithm. because it can be easily modified to apply our three ideas to effectively model our wanted dynamics and still keep the physical interpretation. To obtain the desired behaviour, we have performed the following modifications:

1. Particles in the original algorithm have been substituted by actors.

2. We have revamped the attraction force in a way that now it takes the size, the distance and social value of each actor into account:

$$\mathbf{F}_{xy} = \frac{T((m_x m_y)^c, A_C(x,y))}{m_x^p}, \frac{\mathbf{s}_x - \mathbf{s}_y}{|\mathbf{s}_x - \mathbf{s}_y|^3} dt \tag{3.26}$$

3. The collision condition and fusion procedure for two particles has been adapted to actors.

4. We have replaced the idea of position by the idea of influence. The position matrix has been substituted by an influence matrix, S.

5. Besides the influence matrix, we keep an Affinity matrix, A_C, that contains the affinity for each pair of actors and/or communities alongside the execution of the algorithm.

The publication associated with this results is the following: Fumanal-Idocin, J., Alonso-Betanzos, A., Cordón, O., Bustince, H., & Minárová, M. (2020). Community detection and social network analysis based on the Italian wars of the 15th century. *Future Generation Computer Systems*, 113, 25-40.

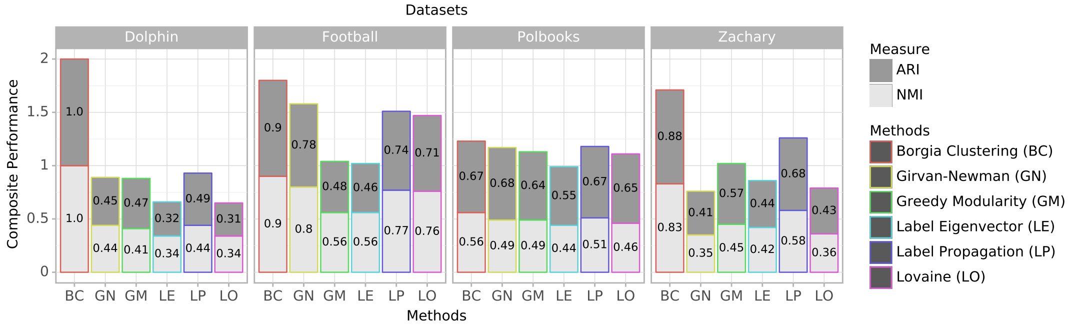


Figure 3.6: Comparison of different Community Detection Algorithms. We have compared the Borgia Clustering algorithm against three modularity optimization methods: Girvan-Newman (74), Newman greedy modularity optimization (75), and the Lovaine algorithm (76). We have also compared it against using eigenvalues of matrices (77) to detect communities and label propagation (78).

3.6 A Generalization of the Sugeno integral to aggregate Interval-valued data: an application to Brain Computer Interface and Social Network Analysis

In this work we propose to extend the concept of interval-valued Sugeno integral given in (79) to a more general Sugeno-like integral, where the min and max operation are substitued by two more general functions, namely F and G, in a similar way as the generalization introduced in (56).

We define our IV-FG-Sugeno like as:

Definition 6 *Let $\preceq$ be an admissible order on $L([0,1])$, $m : 2^N \to L([0,1])$ be an IV fuzzy measure w.r.t. $\preceq$ and $F : L([0,1]) \times L([0,1]) \to L([0,1])$, $G : (L([0,1]))^n \to L([0,1])$ be functions. We say that a triplet (m,F,G) satisfies Condition (WDS) if for all $X_1,\ldots,X_n \in L([0,1])$ and all possible permutations σ_1,σ_2 on N such that $X_{\sigma_1(1)} \preceq \ldots \preceq X_{\sigma_1(n)}$ and $X_{\sigma_2(1)} \preceq \ldots \preceq X_{\sigma_2(n)}$ it holds:*

$$G\Big(F(X_{\sigma_1(1)},m(E_{\sigma_1(1)})),\ldots,F(X_{\sigma_1(n)},m(E_{\sigma_1(n)}))\Big) = G\Big(F(X_{\sigma_2(1)},m(E_{\sigma_2(1)})),\ldots,F(X_{\sigma_2(n)},m(E_{\sigma_2(n)}))\Big), \quad (3.27)$$

where $E_{\sigma_j(i)} = \{\sigma_j(i),\ldots,\sigma_j(n)\}$ for $j \in \{1,2\}$.

Definition 7 *Let n be a positive integer, $\preceq$ be an admissible order on $L([0,1])$ and let a triplet (m,F,G) satisfies Condition (WDS). An interval-valued Sugeno-like FG-functional with respect to m is a function $\mathbf{S}_m^{F,G} : (L([0,1]))^n \to L([0,1])$ given by*

$$\mathbf{S}_m^{F,G}(X_1,\ldots,X_n) = G\Big(F(X_{\sigma(1)},m(E_{\sigma(1)})),\ldots,F(X_{\sigma(n)},m(E_{\sigma(n)}))\Big) \quad (3.28)$$

for all $X_1,\ldots,X_n \in L([0,1])$, where σ is a permutation on N such that $X_{\sigma(1)} \preceq \ldots \preceq X_{\sigma(n)}$ and $E_{\sigma(i)} = \{\sigma(i),\ldots,\sigma(n)\}$.

Depending on the F, G functions chosen, this function can show different properties. We have studied idempotency, internality, positive and min-homogeneity, comonotone maxitivity and comonotone minitivity, boundary conditions, monotonicity and property of giving back the fuzzy measure. The proofs for this properties are developed in the associated paper with this results (which can be seen at the end of this section).

We have used this IV-FG-Sugeno like in two different settings:

- MI-based BCI classification framework.
- Social network analysis.

For the case of MI BCI classification, we have used the same bci framework as in our experiments in the previously explained paper in Section 3.3. However, in this case we have used intervals as a mean to express the uncertainty related to each prediction. We have done so by using different kinds of classifiers for each sample. Then, the interval is constructed taking as the lowest bound the lowest value obtained by any of the classifiers, and the same thing for the upper bound. Since we do this for four different wave bands, we obtained four different intervals to aggregate. We do so using the proposed FG Sugeno-like integral. Finally, this results in a final interval-valued logit for each class. We choose the highest interval according to an admissible order.

For the case of Social network analysis, interval-valued affinity functions (IV-affinity functions) as functions that characterise the relationship between two actors, x,y with an interval in the $[0,1]$ range, where the width of that interval represents the difference in commitment between two parties, contrsucted using a numerical affinity function (80). We construct an interval-valued affinity function using a previously computed numerical affinity function. Then, the interval is constructed as:

$$F_{C,IV}(x,y) = [\min\{F_C(x,y), F_C(y,x)\}, \max\{F_C(x,y), F_C(y,x)\}] \tag{3.29}$$

Because $F_C(x,y) \neq F_C(y,x)$ in most relationships, this means that in most of the actors give different levels of commitment than their counterparts. In real life, these kind of situations are usually solved by finding a compromise between both parties. The IV-affinity function models this idea, representing the actual relationship that it is formed with an interval that ranges from both levels of commitment. The interval models the fact that we know that the final compromise achieved by both actors should be between both commitment levels, but we do not know exactly where.

One of the main difference between IV-affinity functions and their numerical counterpart is that they are symmetric. This can be convenient, as it allows to represent the relationship between two parties with one interval instead of two numerical values. This also opens the possibility of using some of the existing methods that require symmetric matrices in social network analysis (13, 81), while retaining the desirable properties of affinity functions, i.e. zeros-sum game philosophy or local-only interactions taken into account (82).

Using IV-affinities and our the FG-Sugeno like integral, we propose a series centrality measures to characterise the tendency of each actor to form relationships that have very different levels of commitment, and if the actor tends to show more or less commitment than the other party in each of its relationships.

The proposed centrality measures are:

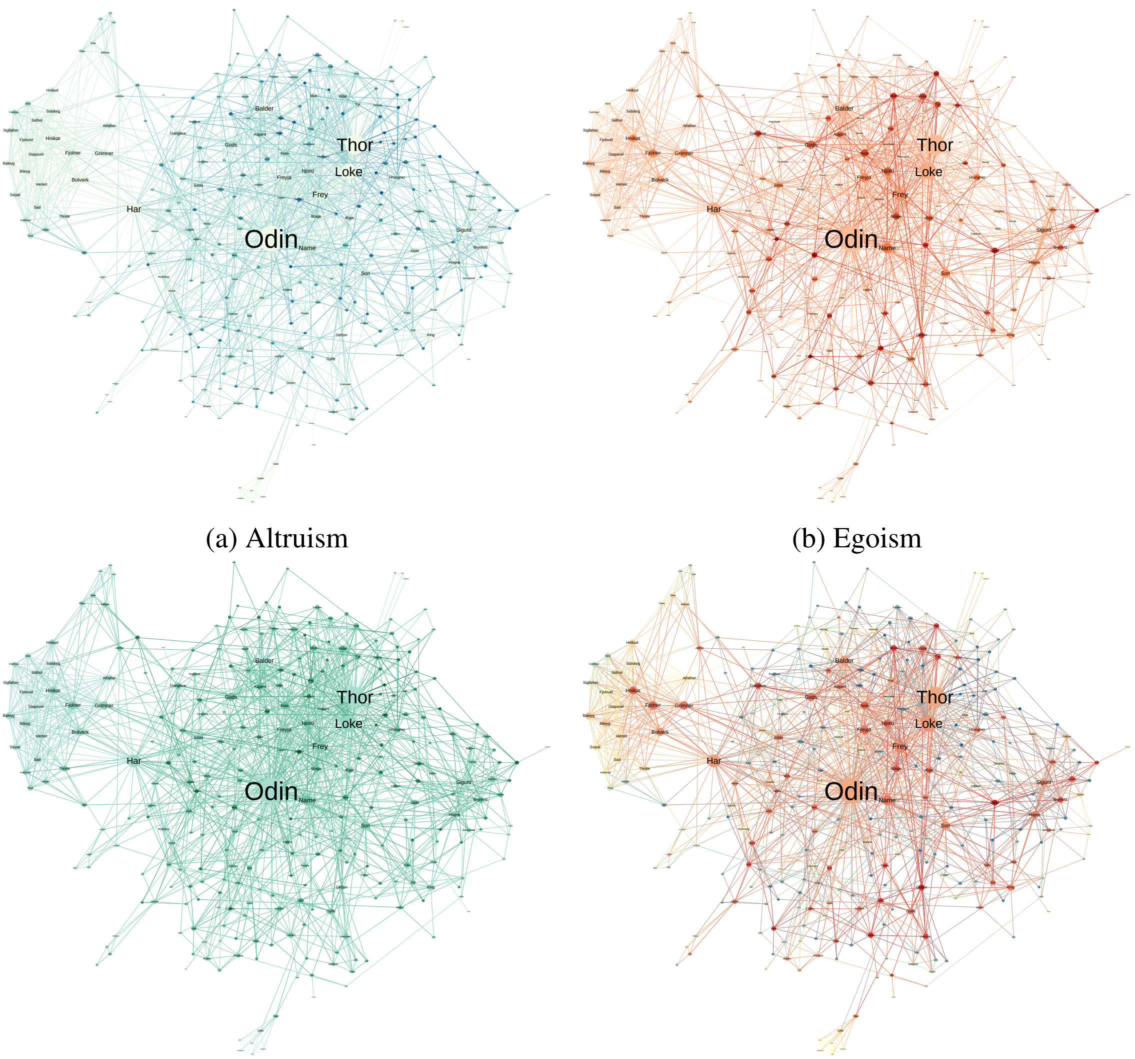
1. **Asymmetry** is the tendency of the actor to form relationships with different levels of commitment.

2. **Altruism** is the tendency of the actor to form relationships in which its level of commitment is bigger than the other party.

3. **Egoism** is the tendency of the actor to form relationships in which its level of commitment is lesser than the other party.

4. **Generosity** is the difference between altruism and egoism. A positive generosity means that overall, the actor tends to give more commitment in its relationships than the other part. A negative generosity means that the actor tends to give less commitment than the other part in a relationship.

We have studied the proposed centrality measures in a word association network, constructed using *The Younger Edda* book. *The Younger Edda* is Old Norse textbook of mythical texts, written approximately in 1220 by Snorri Sturluson. This book contains the tales of popular characters in the Nordic folklore, like Odin, Thor or Loki. Figure 3.7 shows the results in this network for asymmetry, egoism, altruism and generosity, colouring each node according the value of each metric.



(a) Altruism

(b) Egoism

(c) Asymmetry

(d) Generosity

Figure 3.7: Altruism (a), Egoism (b), Asymmetry (c) and Generosity (d) for the *Younger Edda* network, marked with different colors. Node size is proportional to the node degree. We can see that most actors are more egoist than altruist. In fact, some of the most important actors in this network show no altruism at all. However, no altruism does not necessarily imply a high egoism. For example, “Odin” and “Thor” show no altruism (so all their in-affinities are higher or equal than their out-affinities), but their asymmetry value is also low.

3.7 Quantifying External Information in Social Network Analysis: an Application to Comparative Mythology

Social network analysis uses centrality measures to ponder the importance of each actor according to a structural property of the network. In this work we present an extension of this concept to take into account external information from the original domain of the network.

We start by giving the notion of Semantic value, as the union of the intrinsic value, ᛦ(x), and the extrinsic value, ᛘ(x):

$$\text{ᛯ}(x) = \cup(\text{ᛦ}(x), \text{ᛘ}(x)). \tag{3.30}$$

Note that this notation reinforces this idea visually in its more general conception: if we superpose the ᛦ(x) and the ᛘ(x) symbols, the result is ᛯ(x).

We define the intrinsic value of an actor x as the inherent properties that this actors possesses independently of the network structure. This concept is quite abstract, and its quantification depends on the specific problem to solve.

The extrinsic value is defined as the union of all the semantic values that actor x neighbours':

$$\text{ᛘ}(x) = \bigcup_{i=1}^{a}\{F_C(X_i,x)\text{ᛯ}(X_i) - \cup_{j\in J_i}\{\cap(F_C(X_i,x)\text{ᛯ}(X_i), F_C(X_j,x)\text{ᛯ}(X_j))\} \tag{3.31}$$

where $J_i = \{j \in \{1,\ldots,a\}, i \neq j\}$. With this expression, we establish that the extrinsic value is the union of the received semantic values from the rest of the actors. Each actor in X sends its own semantic value to x, modulated in each case by the affinity for that relationship, and erasing the redundancies from other relationships. Since the semantic value has a recursive definition, the result of Eq. (3.30) is a set of intrinsic values.

However, if we quantify the intrinsic value as a number, which is convenient in many applications, the resulting value from Eq. (3.30) is also a number, computed with the analogous expression:

$$E(x) = \sum_{i=1}^{a} \max\left(F_C(X_i,x)I(X_i) - \sum_{j\in J} F_C(X_i,X_j)I(X_i)F_C(X_j,x), 0\right) \tag{3.32}$$

denoting the numerical intrinsic value as I. Finally, the numerical semantic value results in:

$$S(x) = I(x) + E(x). \tag{3.33}$$

We can use this as a centrality measure that takes into account both the external information, expressed in the I values, and its propagation through the network, captured through the extrinsic value E.

Another way of using the semantic value is to compare different actors, by means of affinity functions. To do so, we define the concept of semantic affinity.

The semantic affinity of two actors x and y measures the affinity between them based on the idea of how notably we need to change $S(x)$ to convert it to into $S(y)$. Terms that are similar in meaning should have high values of semantic affinity and non-related terms should have a very low semantic affinity. For example, the semantic affinity between “water” and “ice” should be high because these are very close terms and in real life we only need to freeze water below 0ºC to obtain ice. However, the semantic affinity between “water” and “earth” should be lower, as the difference in real life between those concepts is higher.

In order to compute the semantic affinity between two concepts, we propose a new algorithm, called the Pipe algorithm. The Pipe algorithm computes the semantic affinity based on the idea of modelling $S(x)$ as a liquid we need to carry from x to y. Each actor x has a capacity equal to its own semantic value $S(x)$ and each edge $x \rightarrow y$ can carry up to $F_C(x,y) \cdot S(y)$ of that liquid. So, each edge is treated as a “pipe” where the liquid goes and each actor as a bifurcation in the path. Then, we need to carry all the liquid from the source actor to the destination actor using the best possible path. To compute the final semantic affinity value, we will take into account three different aspects of this transportation process: how “good” were the taken paths, the difference in magnitude between $S(x)$ and $S(y)$ and the average affinity values of the emisor.

As a practical case of this methods, we propose to use the semantic value and the semantic affinity in comparative mythology (83, 84, 85). We show the networks obtained for individual mythologies, and the resulting one when we fuse all of them in Figure 3.8. The results of our pipe algorithm are shown in Figure 3.9. There, we showed how this analysis reveals some important interaction between Nordic and Celtic mythologies, and how between the main gods of each mythology and concepts realted to kingship.

The publication associated with these results is the following one: Fumanal-Idocin J., Cordón O., Dimuro G., del-Hierro AR. & Butince H. (2022) Quantifying External Information in Social Network Analysis: an Application to Comparative Mythology, *IEEE Transactions on Cybernetics*.

3.8 The Krypteia ensemble: designing classifier ensembles using an ancient Spartan military tradition

Following a similar methodology as the one that we followed in (80), in this work we present a real historical costume applied to solve a computational problem. In this case, we show how we can train a classifier ensemble following the general ideas of the Spartan Krypteia.

The Krypteia was a particularly brutal initiation rite in the Spartan society for the young men of the higher ranks of the state (87). According to Plutarch (88), each year the young noblemen of Sparta would declare war to the Helot population of Sparta, so that any killing or robbery committed was not considered crime. Armed with nothing but a knife, the young Spartans were left alone and sent out in the night to the Helot settlements. They were supposed to obtain their own methods of survival by stealing and killing in their circumvent area.

The Krypteia ensemble is a novel classifier ensemble algorithm designed to maximise the effectiveness of each individual subject and the variability and performance of their outputs when combined. It consists of three main steps (Figure 3.10):

1. Survival ordeal: in this phase we train each classifier individually. In order to induce diversity in the training process, we modify the training task for each one in a stochastic process, so that some of them have a easier or harder task than the original classification task. We discard all the classifiers that did not meet the expected accuracy rate in their own task.

2. Social ordeal: this second phase follows an OCS scheme in which we discard those classifiers that have a very similar output to other classifiers with higher accuracy rate. The aim is to avoid having samples with much more weak learners classifying them correctly than others.

3. Aggregation learning: we learn which is the most appropriate function to combine the output of all the surviving classifiers.

In order to guarantee some diversity between the different subjects, they are generated using a stochastic process. The idea is that these subjects are generated in batches, and their parameters are obtained from a distribution i.e. the base classifier, the kind of ordeal performed, the bias induced in their predictions. A set of subjects obtained from the same set of distributions is called a Krypteia unit. If we want to obtain a bigger and more diverse ensemble, we can use different Krypteia units, instead of using a big one. The fusion of different Krypteia units is called a Krypteia division. In the same way, fusing different Krypteia divisions is denoted as Krypteia Army.

Stacking these three different levels of complexity allow us to gain some flexibility in the aggregation process. We have opted to study one different aggregation function in each complexity level, which performs better than using the same one in all levels (something that we already noted in our results in (89)).

We compared the performance of our method in a collection of datasets with a list of different ensemble design approaches, including Random Forest, Gradient Boosting, using meta features and other classical approaches:

- Adaboost (90): it serially trains each classifier. In each iteration, it weights each instance according to its difficulty to be classified, aiming to correctly classify it in the next iteration. For our experimentation, we have used 50 decision trees to form the Adaboost.

- Bagging (Bootstrap Aggregation) (91): it aims to increase accuracy by combining the outputs of the classifiers in the ensemble that were trained using different subsamples of the original data. Sampling with replacement is used to train all the classifiers in the ensemble and thus some of the instances may appear more than once in the training set. For our experimentation, we have used 10 decision trees to form the Bagging classifier.

- Majority vote SVM: it consists of different SVM classifiers trained with a different kernel. For our case, we have trained 5 different RBF kernels classifiers with five different σ parameters evenly spaced such as: $[0.5, 1.5]/(number of features)$.

- Random Forest (92): it combines the output of many different decision trees computed from different subsamples of the original data. The final decision is taken as the majority vote of all of the trees outputs'. We have set as 100 the number of trees in the Random Forest for our experimentation.

- K-Nearest Oracles Eliminate (K-NORAE) (93): it selects the classifiers that correctly classify all the samples in their region of competence.

- Dynamic Ensemble Selection Multiclass Imbalance (DES-MI) (94): it generates artificial training sets of randomly balanced data and then chooses the classifiers that correctly discriminated the minority class samples.

- Randomized Reference Classifier (DES-RRC) (95): it combines Dynamic Ensemble Selection with a measure to evaluate each possible classifier in the final ensemble using a reference classifier.

- META-DES (96): it selects a set of classifiers from a list, using five different meta-features to test each classifier's competence.

- Extreme Gradient Boosting (XGBoost) (97): gradient boosting is a generalization of Adaboost that consist of using a differentiable loss function. This function is optimized using a gradient descent procedure, so that in each step a new weak learner is included to reduce the loss of the system. Gradient boosting is considered to be the state-of-the-art in classification of tabular data (98).

Table 3.6: Average performance for different ensemble classifiers and the best instance of Krypteia ensemble classifiers used.

Algorithm	Accuracy
Adaboost	83.84 ± 19.65
Bagging	90.99 ± 9.81
Majority Vote SVM	68.90 ± 21.84
Random Forest	92.61 ± 8.53
K-NORAE	90.05 ± 11.00
DES-MI	93.87 ± 10.59
DES-RRC	92.74 ± 10.72
META-DES	93.60 ± 10.70
XGBoost	95.14 ± 9.81
SVM-Krypteia Mean	93.30 ± 10.34
Krypteia Unit	94.09 ± 14.36
Krypteia Division	94.49 ± 12.85
Krypteia Army	**95.83 ± 8.35**

As shown in Table 3.6, we found better results using our Krypteia ensembles than most approaches, and the Krypteia Army was found to be the best ensemble of the compared ones.

Specific details about each of these steps can be found in the publication associated with this work:

Fumanal-Idocin J., Cordón O. & Bustince H. (2023) The Krypteia ensemble: designing classifier ensembles using an ancient Spartan military tradition *Information Fusion*

3.9 ARTxAI: Explainable Artificial Intelligence Curates Deep Representation Learning of Artistic Images

In this work we study the generalisation capabilities of a deep learning model used in artistic image classification. We have chosen a ResNet50 model as it is popular in the literature for this kind of task (21, 99, 100, 101). Then, we propose to extract interpretable features from the paintings, which can then be used to obtain explainable patterns using fuzzy rules.

For our experimentation we have used the SemArt and the Wikiart datasets. The SemArt dataset (102). This dataset consists of 21,384 painting images. Following the original data partition in (102), 19,244 images are used for training (i.e. a 90%), 1,069 for validation, and 1,069 for test (i.e. a 5% each). Each painting has associated a textual artistic comment.

In this dataset four different classification tasks are proposed:

- Type: each painting is classified according to 10 different common types of paintings: portrait, landscape, religious, etc.
- School: each painting is identified with different schools of art: Italian, Dutch, French, Spanish, etc. There are a total of 25 classes of this kind.
- Timeframe: The attribute Timeframe, which corresponds to periods of 50 years evenly distributed between 801 and 1900, is used to classify each painting according to its creation date. We consider only the timeframes where at least 10 paintings are present. This corresponds to 18 classes.
- Author: corresponds to the author of each paintings. We consider a total of 350 painters, that comprise the set of authors with more than 10 paintings in the dataset.

The WikiArt dataset contains over 81,000 images of fine-art paintings, representing a wide range of artistic styles and historical periods, from the 11th century to the present day. Each image in the dataset is accompanied by a set of metadata, including the title of the artwork, the artist, the year of creation, the medium used and the dimensions of the artwork, among other attributes.

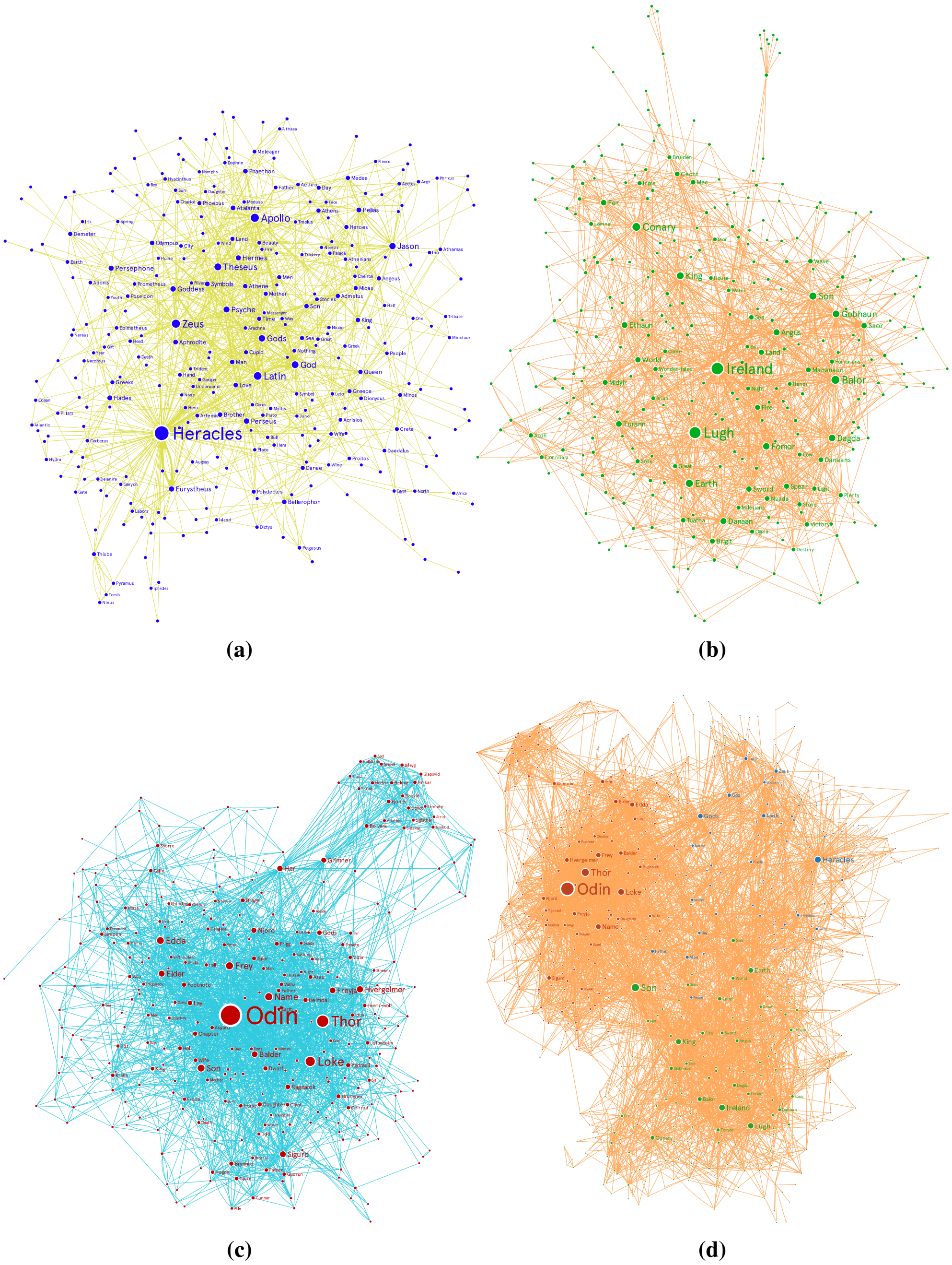
First, we have studied the performance of different models in the SemArt classification tasks. We use models that take into account only visual information and also some that use contextual information by encoding the commentaries for each painting. We have also considered Multi-task models (MTL) that are trained for the four tasks as the same time. We found that the best results were obtained using a combination of both approaches (Table 3.7).

Secondly, we extracted the interpretable features using the grad-CAM algorithm and a style predictor for each image. Grad-CAM is an algorithm used for X-AI in deep learning models that highlights the parts of the image that were relevant in the network prediction. For each image we compute the percentage of the image that was marked as relevant, the sharpness of the boundaries between relevant and non relevant parts of the image and the number of significant blobs detected. On the other hand, we trained a model on the Wikiart dataset to characterise the styles of the SemArt paintings. Combining both approaches we describe each painting according to the scores to these characteristics.

Then, we use this features to fit a Fuzzy Rule Based classifier to obtain explainable rules to match these features to the behaviour of the Resnet model used for classification on the SemArt or to distinguish paintings between two similar artists (Table 3.11).

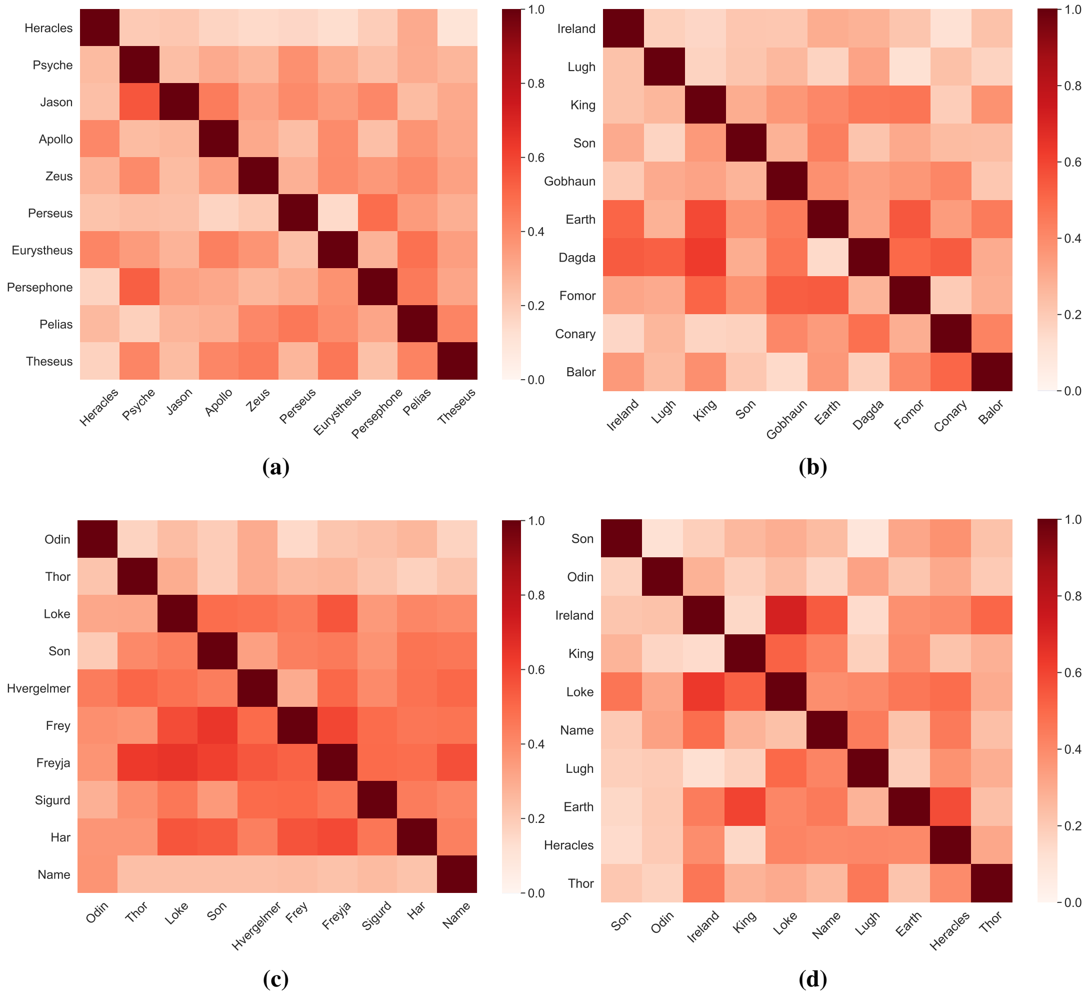
Table 3.7: Correct Classification Ratio results for the different attributes on SemArt Dataset.

Method	Type	School	TimeFrame	Author
VGG16	0.706	0.502	0.418	0.482
ResNet50	0.726	0.557	0.456	0.500
VGG16 MTL	0.732	0.585	0.497	0.513
ResNet50 MTL	0.763	0.565	0.464	0.431
BoW + FCM	0.794	0.655	0.604	0.238
BoW + FCM-apppend	0.802	0.654	0.584	0.230
TF-IDF + FCM	0.786	0.645	0.604	0.229
TF-IDF + FCM-append	0.778	0.654	0.589	0.226
BoW$_{100}$ + FCM	0.792	0.630	0.586	**0.559**
TF-IDF + FRBC	0.785	0.643	0.597	0.233
TF-IDF + FRBC-append*	0.759	0.623	0.533	0.154
CLIP-context	0.784	0.649	0.601	0.215
MTL-FCM	**0.804**	**0.691**	0.618	0.531
MTL-CLIP	0.790	0.677	**0.630**	0.551



(a) (b) (c) (d)

Figure 3.8: Word co-occurrence networks. Each network is formed using the 300 most repeated entities in each corpus. We consider a connection between two words every time they appear less than 10 words apart from each other in one of the analyzed texts. **a.** *Greek Myths* **b.** *Celtic Wonder-Tales* **c.** *The Younger Edda* **d.** Fusion network of the three cultures. Node size is directly proportional to the in-degree measure and the layout algorithm considered is Force Atlas 2 (86).



(a)

(b)

(c)

(d)

Figure 3.9: Semantic affinities in all the networks studied. We chose the 10 most repeated entities in each text to compare themselves. **a** *Greek myths* network. **b** *Celtic Wonder-Tales* network. **c** *Younger Edda* network. **d** Fused myths network.

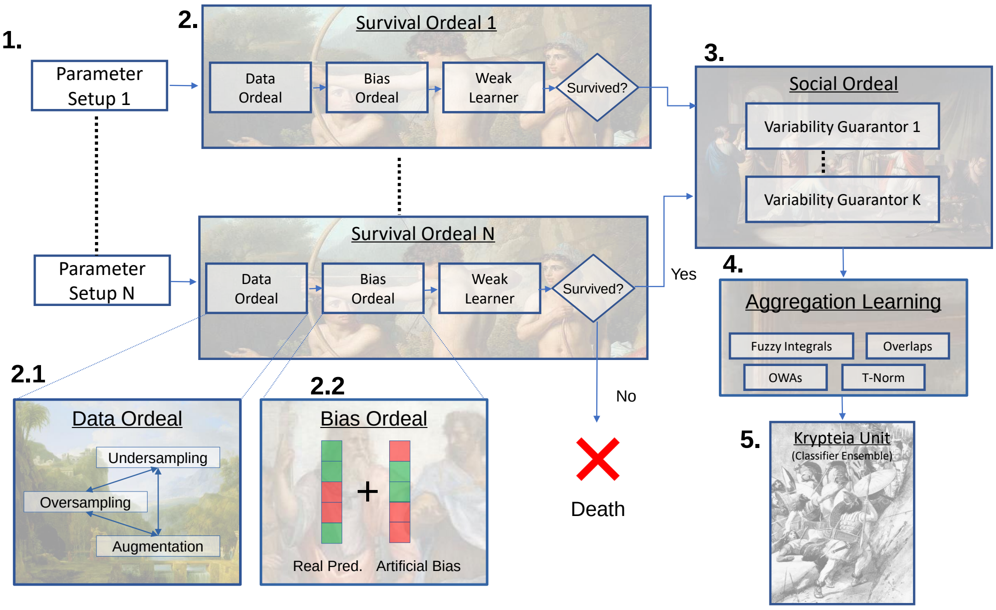


Figure 3.10: Visual scheme of the Krypteia algorithm. **1.** We generate random sets of parameters. **2.** Each of these parameter settings creates a different Survival ordeal, where the weak learners need to correctly solve a stochastically modified version of the original classification task. The modification includes the Data ordeal (**2.1**), where we manipulate the training data, and the Bias ordeal (**2.2**), where we compute a random vector that we will add as an artificial bias to each of the classifiers predictions. **3.** If they pass the Survival ordeal, a selection of the surviving classifiers is performed by *k* Variability Guarantors in the Social ordeal to minimize redundancies. **4.** We choose the best aggregation function for the decision making phase of the ensemble. **5.** The output of the Social ordeal is the Krypteia Unit.

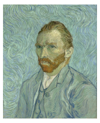
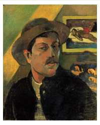
Author		Antecedents	DS	Train Acc	Test Acc
	1	IF New Realism IS Low AND Post Impressionism IS Medium	0.0076	0.5000	0.0000
	2	**IF Early Renaissance IS Medium AND New Realism IS Medium AND Synthetic Cubism IS Medium**	0.0740	0.7777	1.0000
	3	**IF Early Renaissance IS Low AND Synthetic Cubism IS High**	0.2517	0.9390	0.8888
	4	**IF Synthetic Cubism IS Low**	0.4624	0.9097	0.9250
	5	IF Contemporary Realism IS Medium AND Synthetic Cubism IS Low AND Relevant area IS Low	0.0092	0.0000	0.0000
	6	**IF Contemporary Realism IS Medium AND Minimalism IS Low**	0.3389	0.7586	0.7692
	7	IF Early Renaissance IS Medium AND Minimalism IS Medium AND Synthetic Cubism IS Medium	0.0124	0.0000	0.0000

Figure 3.11: Rules that differentiate Van Gogh from Gauguin paintings.

3. DISCUSSION OF RESEARCH FINDINGS

4

Conclusions

La verdad padece, pero no perece.

Santa Teresa de Jesús

In this dissertation we have proposed new means of fusing information using aggregation functions for both numerical and interval-valued data, and we have also developed a novel kind of functions called the affinity functions for social network analysis. Based on these new functions, we have developed the following algorithms and frameworks, designed to tackle a wide range of problems:

- Enhanced multimodal fusion framework for BCI signal classification.
- Two variants of an interval-valued Multimodal Fusion framework for BCI signal classification.
- Affinity functions and the Borgia clustering algorithm to perform community detection in social network analysis.
- Semantic value and semantic affinity to ponder and compare actors in a social network that we applied to comparative mythology.
- Krypteia ensemble to perform classification in tabular data.

As a result of the theoretical and practical results of this dissertation, we have arrived at the following conclusions:

- The choice of different aggregation functions can have a meaningful impact in the final performance of a classification system, both in signal classification or standard tabular data. Some of them can have additional properties in terms of complexity and computational time, which should be taken into account in order to choose one. For example, Choquet and Sugeno integrals require a fuzzy measure, and some T-norms are more expensive to compute than others because of their recursive nature.

- Multi-cost aggregation choosing functions and interval-valued moderate deviations alleviate the problem of choosing the aggregation function. Choosing a cost function is usually easier and very common in many optimization problems.

- BCI systems heavily rely on the features used the preprocessing used and temporal series analysis can help alleviate the effects of possible artifacts in the EEG measurements. Besides, choosing the correct aggregation function that models the interactions and redundancies between features can be very effective to improve the quality of the system.

- Affinity functions represent the “local view” that one actor has over the whole network. This favours the success of algorithms that exploit this but is prejudicial to those that require a global view of the system. They can be combined using aggregations to obtain functions that combine the different social traits that they model. This is helpful in the Borgia Clustering to take into account different social aspects when joining actors into a community.

- External information can be propagated using affinity functions in a network, acting as a fusion function, through the semantic value and semantic affinity concepts. This requires a modelization of this external value into a numerical representation. When this information comes from the original material of the network, additional conclusions can be reached from the original source. When this information is extracted using external resources i.e. sentiment analysis, additional conclusions can be reached, not possible using only the original information.

5

Future Lines

Avancez courageusement. Peur de rien.
Confiance en Dieu; tout ira bien.
Sainte Jeanne d'Arc

Fides ad ea quae non videntur, spes vero
ad ea quae non sunt.
Santo Tommaso d'Aquino

As the number of findings and results reported in this dissertation is quite wide, the range of possible future lines comprises different application and theoretical possibilities. In the following, we introduce the ones that we think are more promising.

Extension of Affinity functions with reinforcement learning

Affinity functions are proposed as a means to measure an actor's "world" view, using information that is considered to be visible from that actor perspective. Multi-Agent Reinforcement Learning systems are systems where agents learn to cooperate to solve a collective task. One of the problems of these kinds of system is to measure the real communication of the systems and how it is affected by the amount of information available to each actor. Affinity functions could be used to model the local information for each actor and to make them communicate in ways that are plausible in the real world.

Extending the Theory and Operators that Work with Affinity Functions

Affinity functions were defined in the most general way according to their initial idea and intentionality. Some additional conditions could be required in order to extend common used operators and transformations used in adjacency networks. For example, as symmetry is recovered using IV-affinity functions, these kinds of affinity functions could be the basis to extend the concept to such operators.

Escaling of the proposed algorithms

In this dissertation we have introduced the:

- Enhanced Multimodal Fusion framework for BCI signal classification.
- Borgia Clustering for community detection.
- Gravity clustering for anomaly detection.
- Krypteia ensemble for tabular data classification.

All of these algorithms worked well when the size of the data was small. However, some of the most interesting problems in artificial intelligence require large amounts of data, which requires the used of batches, distributed computing, etc. All of these algorithms offer possibilities in terms of implementation and further development to adapt them to higher volumes of data.

Extending comparative mythology

One of the key applications of the results in this dissertation was in comparative mythology. This study offered satisfactory results, but could be expanded by adding more textual resources and other types of information. We also consider to add expert knowledge in some parts of the process in order to obtain a system that can better explain its results.

Understanding visual arts through AI We studied that models used to classify artistic images lacks the capabilities of a human person to understand its context and symbolic information. We could expand the capabilities of existing methods in this task by studying the problem of the semantic affinity in this context. We could also study how Large Language Models perceive fiction texts compared to non-fiction and use this insight in the visual arts.

6

Español: Resumen y conclusiones

6.1 Resumen

La fusión de información es un aspecto crucial del análisis moderno de datos y la toma de decisiones. Implica la integración de múltiples fuentes de información para obtener una comprensión más completa y precisa de un tema determinado. Este proceso es especialmente importante en campos como la informática, la ingeniería y las ciencias naturales, donde se generan grandes cantidades de datos procedentes de diversas fuentes que deben sintetizarse para tomar decisiones con conocimiento de causa. La fusión de información también es esencial en el diseño y la implantación de sistemas inteligentes, ya que permite integrar diversos sensores y fuentes de datos para hacer predicciones y recomendaciones más precisas.

Desde un punto de vista matemático, una forma de estudiar este problema es a través de la idea de funciones de fusión, que toman como entrada un vector de números y devuelven uno solo, representativo de ellos. Un tipo relevante de función de fusión es la familia de funciones de agregación. Estas funciones mantienen dos condiciones de contorno y monotonicidad con respecto a las entradas, que inducen algunas propiedades deseables a la salida de la función. Sin embargo, la fusión de información en los sistemas aplicados comprende algo más que esta noción teórica. A medida que la heterogeneidad, la estructura y el volumen de los datos adquieren mayor relevancia, han surgido otros enfoques para abordar este problema. Por ejemplo, en una estructura de red, las distintas entradas se asocian entre sí según un conjunto preestablecido de relaciones; en las series temporales, los datos presentan dependencias temporales. Cuando se trata de datos no estructurados, como texto, audio e imagen, los enfoques de aprendizaje profundo han tenido mucho éxito en la transformación de este tipo de datos en representaciones vectoriales de números reales utilizando series de transformaciones afines.

A pesar de los esfuerzos previos en este campo, el problema de combinar eficazmente fuentes de información diversas y heterogéneas, sigue siendo un área de investigación abierta y activa. Esto se debe a los desafíos inherentes a la integración de múltiples fuentes que pueden estar en diferentes formatos y pueden tener información contradictoria o incompleta. Por ejemplo, el modo en que la información medida se relaciona con otras fuentes de datos y la fiabilidad de esas medidas dependen en gran medida del procedimiento de medición. De hecho, los sistemas que fusionan la información de esas distintas fuentes presentarán también complejidades adicionales al tener en cuenta las particularidades de cada característica considerada.

En esta tesis, proponemos un conjunto de funciones y algoritmos para tener en cuenta las posibles interacciones, heterogeneidades e incertidumbres cuando se trabaja con distintas fuentes de información. Lo hacemos mediante la teoría de agregaciónes y el análisis de redes sociales, y nos centramos especialmente en aquellos casos en los que los enfoques de aprendizaje profundo no tienen tanto éxito. Aplicamos estos resultados a una amplia gama de problemas, incluyendo la clasificación de señales de interfaz cerebro-ordenador, la clasificación de datos tabulares estándar y la detección de anomalías.

6.2 Conclusiones

En esta tesis hemos propuesto nuevos medios para fusionar información utilizando funciones de agregación tanto para datos numéricos como para datos valorados en intervalos, y también hemos desarrollado un novedoso tipo de funciones denominadas funciones de afinidad para el análisis de redes sociales. Basándonos en estas nuevas funciones, hemos desarrollado los siguientes algoritmos y marcos, diseñados para abordar una amplia gama de problemas:

- Marco de fusión multimodal mejorado para la clasificación de señales BCI. variantes de un marco de fusión multimodal interval-valued para la clasificación de señales BCI.

- Funciones de afinidad y el algoritmo de agrupación de Borgia para realizar la detección de comunidades en el análisis de redes sociales.

- Valor semántico y afinidad semántica para ponderar y comparar actores en una red social que aplicamos a la mitología comparada.

- Krypteia ensemble para realizar clasificación en datos tabulares.

Como resultado de los resultados teóricos y prácticos de esta disertación, hemos llegado a las siguientes conclusiones:

- La elección de diferentes funciones de agregación puede tener un impacto significativo en el rendimiento final de un sistema de clasificación, tanto en clasificación de señales como de datos tabulares estándar. Algunas de ellas pueden tener propiedades adicionales en términos de complejidad y tiempo computacional, que deben tenerse en cuenta para elegir una. Por ejemplo, las integrales de Choquet y Sugeno requieren una medida difusa, y algunas T-normas son más caras de calcular que otras debido a su naturaleza recursiva.

- Las funciones de elección de agregación de costes múltiples y las desviaciones moderadas valoradas en intervalos alivian el problema de elegir la función de agregación. Elegir una función de coste suele ser más fácil y muy común en muchos problemas de optimización.

- Los sistemas BCI dependen en gran medida de las funciones utilizadas el preprocesamiento utilizado y el análisis de series temporales puede ayudar a aliviar los efectos de posibles artefactos en las mediciones de EEG. Además, elegir la función de agregación correcta que modele las interacciones y redundancias entre características puede ser muy eficaz para mejorar la calidad del sistema.

- Las funciones de afinidad representan la "visión local" que un actor tiene sobre toda la red. Esto favorece el éxito de los algoritmos que la explotan, pero perjudica a los que requieren una visión global del sistema. Pueden combinarse mediante agregaciones para obtener funciones que combinen los distintos rasgos sociales que modelan. Esto es útil en el Borgia Clustering para tener en cuenta diferentes aspectos sociales a la hora de unir actores en una comunidad.

- La información externa puede propagarse utilizando funciones de afinidad en una red, actuando como una función de fusión, a través de los conceptos de valor semántico y afinidad semántica. Esto requiere una modelización de este valor externo en una representación numérica. Cuando esta información procede del material original con el que se formó la red, se pueden alcanzar conclusiones adicionales a las que se obtendrían utilizando sólo la información estructural de la red.

Part II

Publications: Published, accepted and submitted works

6.3 Motor-Imagery-Based Brain–Computer Interface Using Signal Derivation and Aggregation Functions

Associated publication:

- Fumanal-Idocin, J., Wang, Y. K., Lin, C. T., Fernández, J., Sanz, J. A., & Bustince, H. (2021). Motor-imagery-based brain-computer interface using signal derivation and aggregation functions. *IEEE Transactions on Cybernetics.*

  Status: Published.

  Impact Factor (JCR 2021): 19.118.

  Categories:

  Computer Science, Artificial Intelligence. Ranking 3/144 (Q1).

  Automation & Control Systems. Ranking 1/65 (Q1).

  Computer Science, Cybernetics. Ranking 1/24 (Q1).

Fumanal-Idocin, J., Wang, Y.-K., Lin, C.-T., Fernandez, J., Sanz, J. A., & Bustince, H. (2022). Motor-imagery-based brain–computer interface using signal derivation and aggregation functions. *IEEE Transactions on Cybernetics*, *52*(8), 7944-7955. https://doi.org/10.1109/TCYB.2021.3073210

6.4 Interval-Valued Aggregation Functions Based on Moderate Deviations Applied to Motor-Imagery-Based Brain Computer Interface

Associated publication:

- Fumanal-Idocin, J., Takáč, Z., Fernández, J., Sanz, J. A., Goyena, H., Lin, C. T., Wang, Y.K., & Bustince, H. (2021). Interval-Valued Aggregation Functions Based on Moderate Deviations Applied to Motor-Imagery-Based Brain–Computer Interface. *IEEE Transactions on Fuzzy Systems, 30(7), 2706-2720.*

  Status: Published.

  Impact Factor (JCR 2021): 12.253.

  Categories:

  Computer Science, Artificial Intelligence. Ranking 11/144 (Q1).

  Engeineering, Electrical & Eletronic. Ranking 10/276 (Q1).

Fumanal-Idocin, J., Takac, Z., Fernandez, J., Sanz, J. A., Goyena, H., Lin, C.-T., Wang, Y.-K., & Bustince, H. (2022). Interval-valued aggregation functions based on moderate deviations applied to motor-imagery-based brain–computer interface. *IEEE Transactions on Fuzzy Systems*, *30*(7), 2706-2720. https://doi.org/10.1109/TFUZZ.2021.3092824

Enlace a Academica-e:

https://hdl.handle.net/2454/42913

6.5 Multi-cost penalty functions applied to Motor-Imagery based Brain-Computer-Interface

Associated publication:

- Fumanal-Idocin, J., Vidaurre, C., Fernandez J., Gomez M. Andreu-Perez J. & Bustince H. (2021). Multi-cost penalty functions applied to Motor-Imagery based Brain-Computer-Interface. *Pattern Recognition*

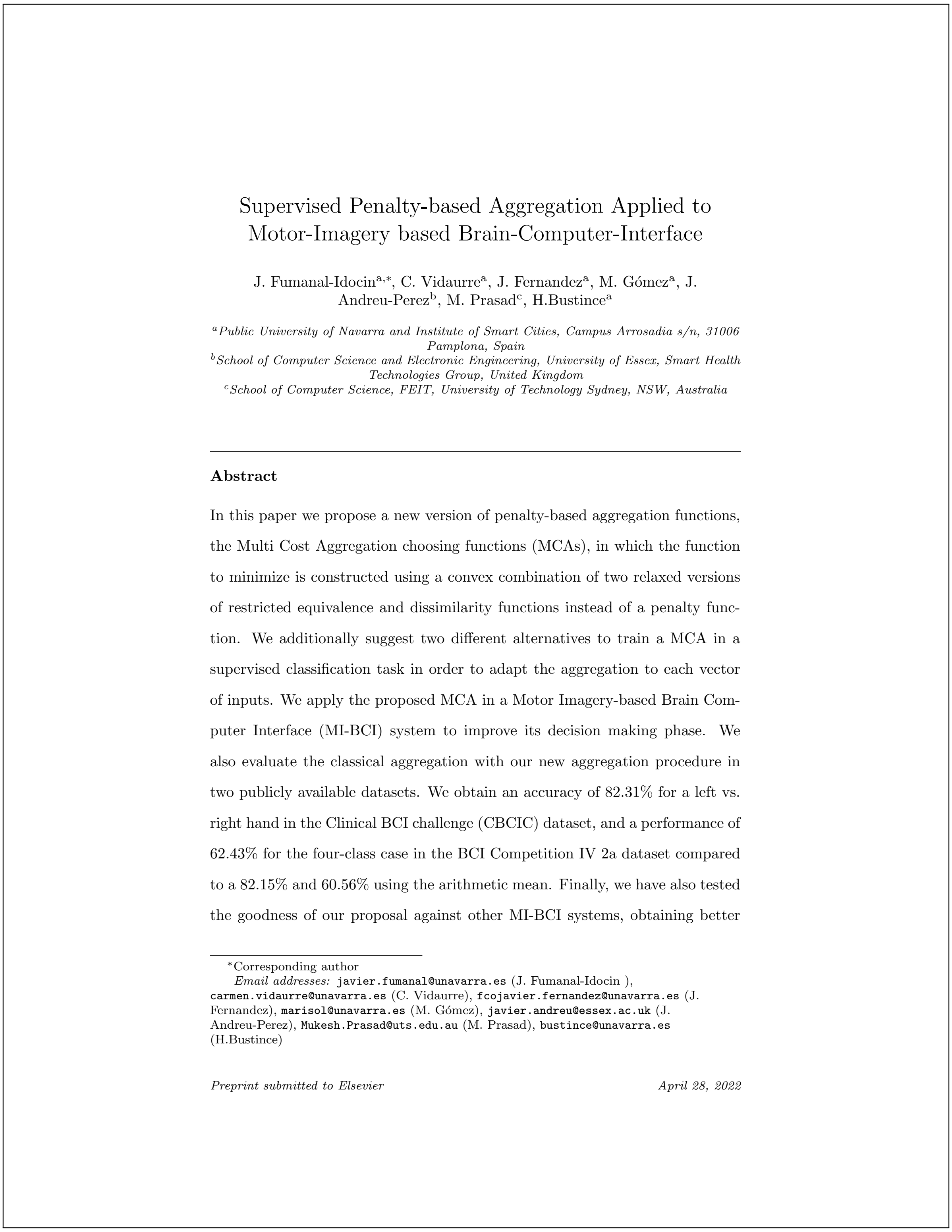
  Status: Submitted.

  Impact Factor (JCR 2021): 8.518.

  Categories:

  Computer Science, Artificial Intelligence. Ranking 22/144 (Q1).

  Engineering, Electrical & Electronic. Ranking 24/246 (Q1).



Supervised Penalty-based Aggregation Applied to Motor-Imagery based Brain-Computer-Interface

J. Fumanal-Idocin[a,*], C. Vidaurre[a], J. Fernandez[a], M. Gómez[a], J. Andreu-Perez[b], M. Prasad[c], H.Bustince[a]

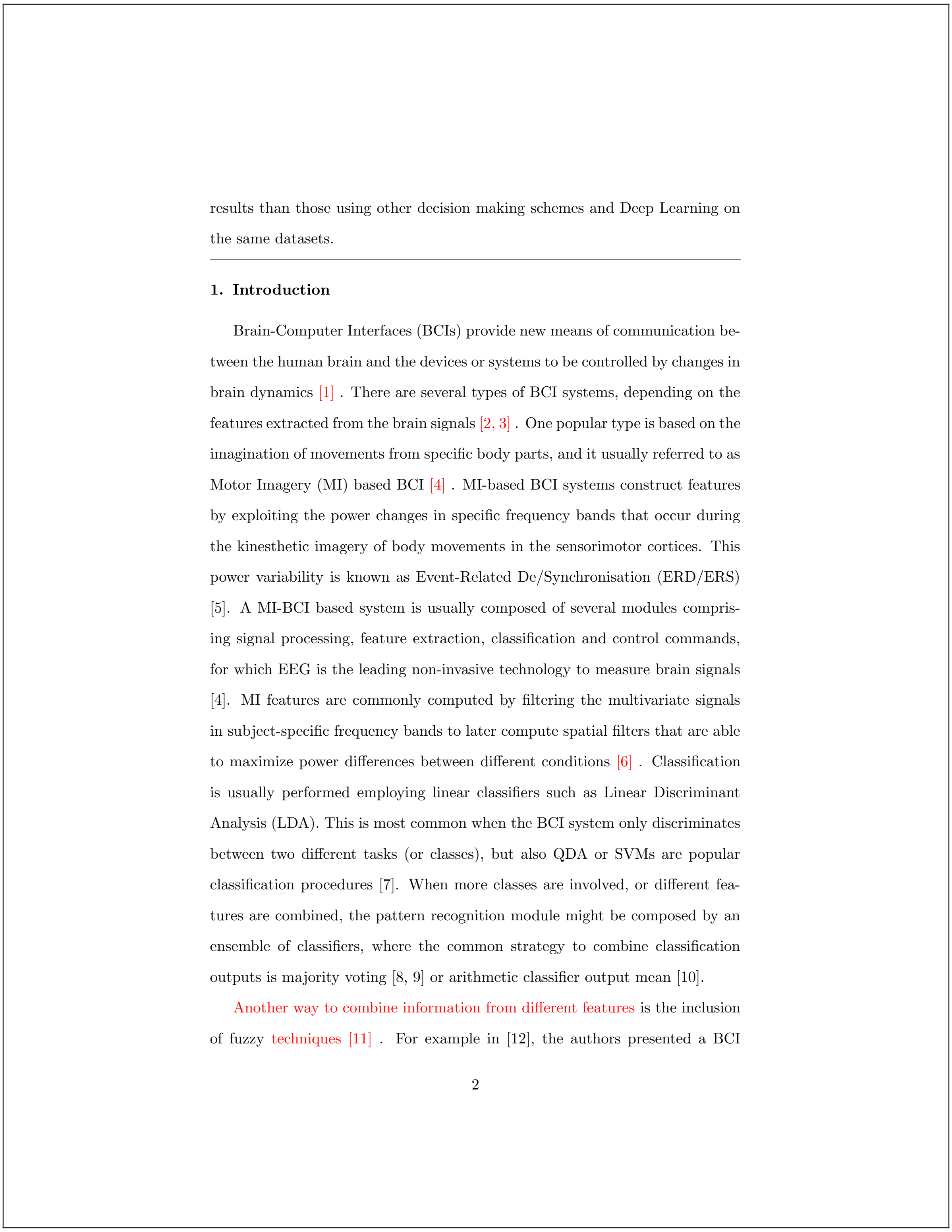
[a] *Public University of Navarra and Institute of Smart Cities, Campus Arrosadia s/n, 31006 Pamplona, Spain*
[b] *School of Computer Science and Electronic Engineering, University of Essex, Smart Health Technologies Group, United Kingdom*
[c] *School of Computer Science, FEIT, University of Technology Sydney, NSW, Australia*

Abstract

In this paper we propose a new version of penalty-based aggregation functions, the Multi Cost Aggregation choosing functions (MCAs), in which the function to minimize is constructed using a convex combination of two relaxed versions of restricted equivalence and dissimilarity functions instead of a penalty function. We additionally suggest two different alternatives to train a MCA in a supervised classification task in order to adapt the aggregation to each vector of inputs. We apply the proposed MCA in a Motor Imagery-based Brain Computer Interface (MI-BCI) system to improve its decision making phase. We also evaluate the classical aggregation with our new aggregation procedure in two publicly available datasets. We obtain an accuracy of 82.31% for a left vs. right hand in the Clinical BCI challenge (CBCIC) dataset, and a performance of 62.43% for the four-class case in the BCI Competition IV 2a dataset compared to a 82.15% and 60.56% using the arithmetic mean. Finally, we have also tested the goodness of our proposal against other MI-BCI systems, obtaining better

[*] Corresponding author
Email addresses: javier.fumanal@unavarra.es (J. Fumanal-Idocin), carmen.vidaurre@unavarra.es (C. Vidaurre), fcojavier.fernandez@unavarra.es (J. Fernandez), marisol@unavarra.es (M. Gómez), javier.andreu@essex.ac.uk (J. Andreu-Perez), Mukesh.Prasad@uts.edu.au (M. Prasad), bustince@unavarra.es (H.Bustince)

Preprint submitted to Elsevier *April 28, 2022*



results than those using other decision making schemes and Deep Learning on the same datasets.

1. Introduction

Brain-Computer Interfaces (BCIs) provide new means of communication between the human brain and the devices or systems to be controlled by changes in brain dynamics [1] . There are several types of BCI systems, depending on the features extracted from the brain signals [2, 3] . One popular type is based on the imagination of movements from specific body parts, and it usually referred to as Motor Imagery (MI) based BCI [4] . MI-based BCI systems construct features by exploiting the power changes in specific frequency bands that occur during the kinesthetic imagery of body movements in the sensorimotor cortices. This power variability is known as Event-Related De/Synchronisation (ERD/ERS) [5]. A MI-BCI based system is usually composed of several modules comprising signal processing, feature extraction, classification and control commands, for which EEG is the leading non-invasive technology to measure brain signals [4]. MI features are commonly computed by filtering the multivariate signals in subject-specific frequency bands to later compute spatial filters that are able to maximize power differences between different conditions [6] . Classification is usually performed employing linear classifiers such as Linear Discriminant Analysis (LDA). This is most common when the BCI system only discriminates between two different tasks (or classes), but also QDA or SVMs are popular classification procedures [7]. When more classes are involved, or different features are combined, the pattern recognition module might be composed by an ensemble of classifiers, where the common strategy to combine classification outputs is majority voting [8, 9] or arithmetic classifier output mean [10].

Another way to combine information from different features is the inclusion of fuzzy techniques [11] . For example in [12], the authors presented a BCI framework employing fuzzy integrals [13] to model classifier interactions. Another example is [14], where the authors proposed the use of interval-valued aggregation functions. Furthermore, the promising results in [12] show that the classifier fusion in the control command phase is crucial to increase BCI performance. However, choosing the best aggregation function for such system depends on several factors, such as the type or number of classifiers used. Based on the theory of aggregation functions [15], one possible method to combine classifier outputs is to use a dissimilarity measure between the data and the fused value. A way of measuring this dissimilarity is the so-called penalty functions.

Penalty functions are defined as a measure of deviation from a consensus value, or in other words, as a penalty for not reaching consensus. They have been widely studied in the fuzzy learning field [16] . Penalty functions can be used to build fusion functions which take into account the lack of similarity between inputs. These functions are called penalty-based functions. Some examples are the weighted arithmetic and geometric means and median.

Penalty functions allow the choice of the "best" possible aggregation according to a dissimilarity measure, thereby solving the problem of choosing an aggregation function for a specific problem. However, care needs to be taken with their design. For example, when the quadratic error is set as a penalty function, the arithmetic mean will be selected as the best possible aggregation regardless of the data to be aggregated [16]. This is due to, by definition, the arithmetic mean of the input values is always the value that minimizes the penalty.

The main goal of this paper is to propose and apply a new method to fuse BCI classification outputs to generate a control command. This method is based on a special type of penalty-based aggregation functions: the Multi-Cost Aggregation-Choosing functions (MCAs). MCAs are similar to penalty-based aggregation functions because they establish a disagreement measure between the original data and the aggregated output in order to determine the "best" aggregation. The disagreement measure is constructed using a convex combination of two cost functions. Depending on the convex combination parameter, the proposed functions are able to obtain more meaningful results regarding which aggregation function is denoted as the "best", than the classical approaches. A second goal is to demonstrate the usefulness of MCA functions to classify MI-based BCI data in comparison to the arithmetic mean or the classical penalty-based aggregation functions. To show that this is the case, we perform several favorable comparisons between different aggregation functions and to other previously published work on the same dataset [12, 17].

The paper is organized as follows. Section 2 revises the concepts of aggregation and penalty functions. Section 3 introduces the main contributions for this work: section 3.1 illustrates the BCI framework, and section 3.2 shows how to process the EEG data. Section 3.3 explains the concept of Quasi-Restricted Equivalence Functions and Quasi-Restricted Dissimilarity Functions and Section 3.4 explains how to use them to construct multi-cost functions. Section 3.5 explains how to mix the different cost functions in a MCA in order to optimize the performance in a supervised learning task and, subsequently, section 3.6 describes how to apply these functions to the BCI MI framework. Section 4 displays the experimental results for the popular BCI IV competition dataset [18] and the Clinical Brain-Computer Interface Challenge (CBCIC) at the IEEE World Congress of Computational Intelligence (WCCI) 2020 [19] using the MCA functions; and in Section 5 we compare those results with other BCI frameworks. Finally, in Section 6 we give our final conclusions and remarks for this work.

2. Preliminaries

This section discusses some of the basic concepts regarding aggregation functions and more precisely, penalty-based aggregation functions.

2.1. Aggregation Functions

Aggregation functions are used to fuse information from n sources into one single output [13]. A function A: $[0,1]^n \to [0,1]$ is said to be a n-ary aggregation function if the following conditions hold for any vectors $(x_1,\ldots,x_n) \in [0,1]^n$:

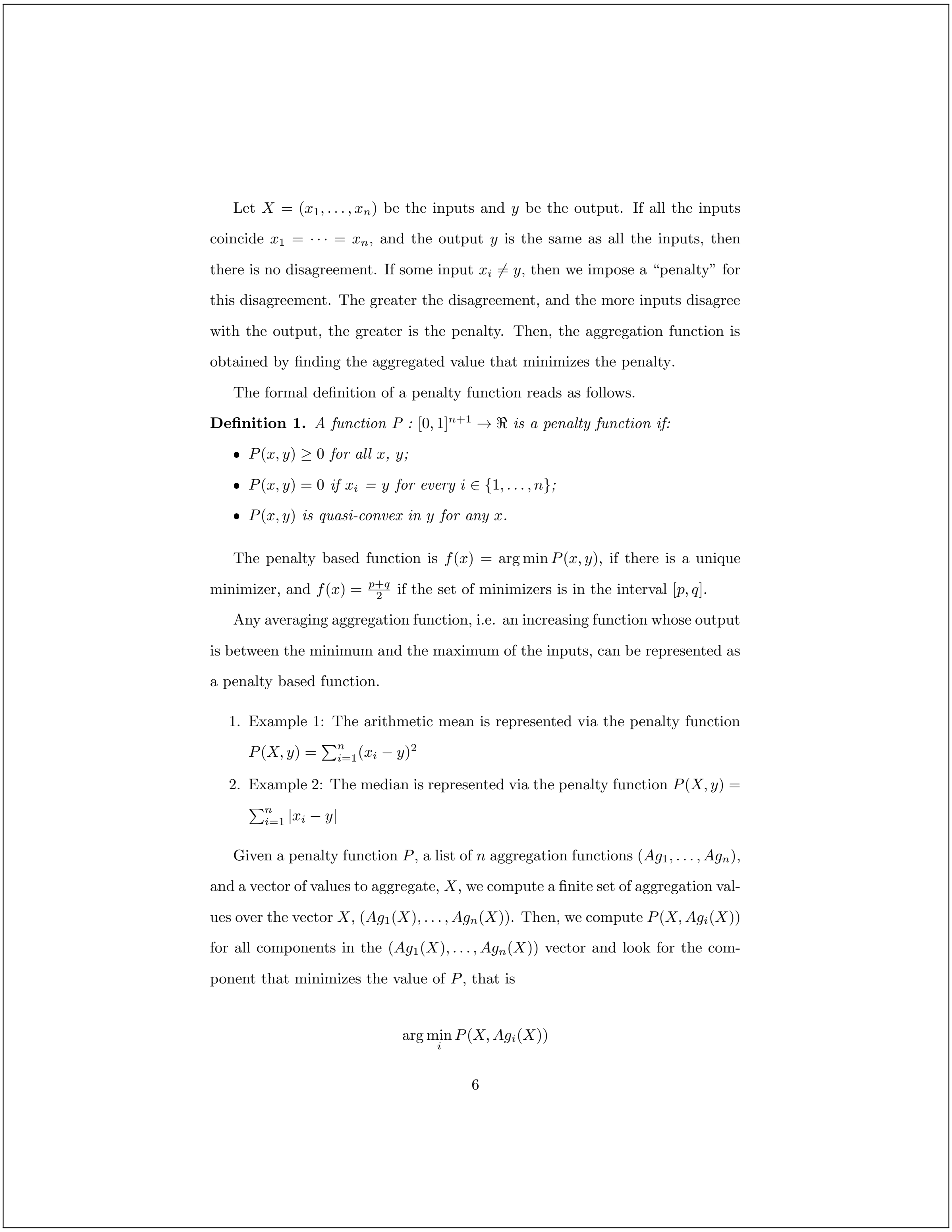
- A is increasing in each argument; that is, for every $x_i \in \{1,\ldots,n\}$, if $x_i < y, A(x_1,\ldots.,x_i,\ldots x_n) \leq A(x_1,\ldots,y,\ldots x_n)$
- $A(0,\ldots,0) = 0$
- $A(1,\ldots,1) = 1$

Some examples of n-ary aggregation functions are:

- Arithmetic mean: $A(X) = \frac{1}{n}\sum_{i=1}^{n} x_i$.
- Median: $A(X) = x_m,$ where for any permutation $\sigma : \{1,\ldots,n\}$ such that $x_{\sigma(1)} \leq \cdots \leq x_{\sigma(n)}, x_m = x_{\sigma(\frac{n+1}{2})}$, if n is odd, and $x_m = \frac{1}{2}(x_{\sigma(\frac{n}{2})} + x_{\sigma(\frac{n+1}{2})})$ if n is even.
- Max: $A(X) = max(x_1,\ldots,x_n)$.
- Min: $A(X) = min(x_1,\ldots,x_n)$.

2.2. Penalty functions

Penalty-based aggregation functions aim to reduce the disagreement between the input data and the aggregated value in an information fusion process. This process is measured using a disagreement measure called the penalty function.



Let $X = (x_1, \dots, x_n)$ be the inputs and y be the output. If all the inputs coincide $x_1 = \dots = x_n$, and the output y is the same as all the inputs, then there is no disagreement. If some input $x_i \neq y$, then we impose a “penalty” for this disagreement. The greater the disagreement, and the more inputs disagree with the output, the greater is the penalty. Then, the aggregation function is obtained by finding the aggregated value that minimizes the penalty.

The formal definition of a penalty function reads as follows.

Definition 1. *A function $P : [0,1]^{n+1} \to \Re$ is a penalty function if:*

- $P(x,y) \geq 0$ *for all* x, y*;*
- $P(x,y) = 0$ *if* $x_i = y$ *for every* $i \in \{1, \dots, n\}$*;*
- $P(x,y)$ *is quasi-convex in* y *for any* x*.*

The penalty based function is $f(x) = \arg\min P(x,y)$, if there is a unique minimizer, and $f(x) = \frac{p+q}{2}$ if the set of minimizers is in the interval $[p,q]$.

Any averaging aggregation function, i.e. an increasing function whose output is between the minimum and the maximum of the inputs, can be represented as a penalty based function.

1. Example 1: The arithmetic mean is represented via the penalty function $P(X,y) = \sum_{i=1}^{n}(x_i - y)^2$
2. Example 2: The median is represented via the penalty function $P(X,y) = \sum_{i=1}^{n} |x_i - y|$

Given a penalty function P, a list of n aggregation functions $(Ag_1, \dots, Ag_n)$, and a vector of values to aggregate, X, we compute a finite set of aggregation values over the vector X, $(Ag_1(X), \dots, Ag_n(X))$. Then, we compute $P(X, Ag_i(X))$ for all components in the $(Ag_1(X), \dots, Ag_n(X))$ vector and look for the component that minimizes the value of P, that is

$$\arg\min_i P(X, Ag_i(X))$$

3. Methods

This section illustrates the BCI framework used and how the EEG data were processed. We also introduce the new concepts of Quasi-Restricted Equivalence and Quasi-Restricted Dissimilarity Functions(Q-REF and Q-RDF), and how to construct the newly developed MCAs.

3.1. Motor Imagery Brain Computer Interface Framework

The usual modules of a BCI system can be summarized as follows:

1. EEG acquisition with an EEG device, notch filtering to remove power line noise and possibly subsampling and/or bad impedance channel removal.
2. Feature extraction from the EEG data measured. Often, band pass filtering in subject-specific or fixed bands is applied to extract specific EEG oscillations [20]. Then, some dimensionality reduction procedure such as Spatio-Spectral Decomposition (SSD) might be applied [21] . Then, Common Spatial Patterns (CSP) are usually employed to compute optimized spatial filters [22] to separate MI tasks. Other possibilities include using Riemannian geometry [23] or time-domain features modeling the signal as Laplacian and Gaussian random process [24]. The extracted features are log-transformed to normalize them.
3. Pattern classification is performed on the extracted features. In this paper we use an ensemble of classifiers to decode the imagery commands. Each base classifier is trained using for example a different band and the final decision is taken combining all of them. The most common way to obtain the final decision is to compute the arithmetic mean of the outputs of all the base classifiers (each one provides a probability for each class), and take the class with a higher aggregated value. The most common base classifier used in combination with CSP filters and log-transformed power values is the Linear Discriminant Analysis (LDA) [25].

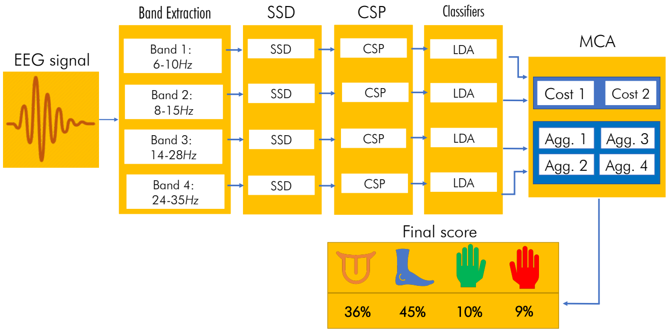


Figure 1: Visual representation of the framework used in this study. First, we measure the EEG band, and extract the information from four different frequency bands. Then, we apply SSD and subsequently CSP to reduce dimensionality and extract features from each band. From each frequency band we train a different LDA classifier. We make a final decision by aggregating the output from all the LDA classifiers using a MCA (detailed in Section 3.4), which results in the estimated probabilities for each one of the possible classes.

A schematic view of the framework used in our experimentation can be found in Fig. 1.

3.2. Feature extraction and classification

In order to extract features, the EEG data were first filtered in four fixed and overlapped frequency bands, covering the range from low μ to high β bands: 6-10, 8-15, 14-28 and 24-35 Hz.

In more detail, the time interval to extract features was optimized for each of the bands using heuristics based on the Event-Related Desynchronization /Synchronization (ERD/ERS) effects typically observed in motor imagery data [26]. The time-resolved ERD/ERS curves were computed as follows: first, the EEG data were spatially filtered using small Laplacian derivations and those channels covering the sensorimotor cortex were selected. Then, these data were band-pass filtered at the band of interest. For each selected Laplacian derivation, the Hilbert transform [27] was applied to obtain the amplitude envelope of the oscillations. EEG activity processed in this way was averaged across epochs separately for each class (left hand/right hand/feet/tongue MI). The time-resolved ERD curve was calculated for each channel according to: ERD = ${}^{100*(\mathrm{POST}-\mathrm{PRE})}/_{\mathrm{PRE}}$, where POST is the EEG processed activity at the post-stimulus interval and PRE is the average activity in the pre-stimulus interval (-500 to 0 ms). Then, the subject-specific time interval (a range of time samples within the active trial time) was selected using heuristics on the ERD/ERS values (see [28]). These heuristics were based on the pair-wise class discriminability of each time sample that was assessed by the signed r^2-value (point biserial correlation coefficient). The signed r^2-value is a correlation coefficient between a real variable (in this case the ERD/ERS value) and a dichotomous one containing class information. Signed r^2-values were computed for each channel and time sample separately and smoothed with a sliding window of 200 ms. The most discriminative time samples were selected using signed r^2-coefficient with 0.8 as threshold value and more samples were iteratively added depending on the averaged discriminative value of the new interval. Figure 2 shows time intervals averaged across subjects and partitions. They mostly cover the period between 1 and 4 seconds during feedback, although they are slightly different depending on the band.

After selecting time intervals for each class pair, the EEG data were epoched to form post-stimulus trials. The total dimensionality of the data was then reduced using SSD on the band of interest [29]. This method allows the extraction of oscillatory neuronal sources with optimized Signal-to-noise ratio. It linearly decomposes multivariate data maximizing the power of the signals at specific bands and at the same time minimizing it at the neighbouring frequency bins. After applying SSD, the selected sources were spatially filtered using common

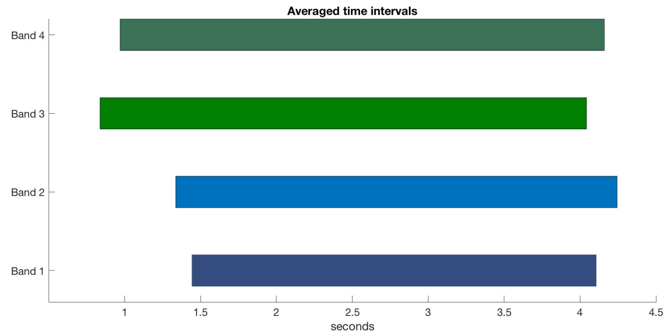


Figure 2: Average time interval chosen for each wave band.

spatial pattern (CSP) analysis [28] . Then, log-variance features were computed for each trial of the training set.

The features of the test set were computed by temporally filtering the EEG data in the four bands of interest. For each band and class pair, the corresponding SSD and CSP spatial filters were then applied. Then, the data were epoched using the previously found time intervals. Finally, the variance and logarithm were applied to each of the features in each trial. The features were then log-transformed and LDA classifiers [30] were trained for subsequent classification. We also considered the use of SVM classifiers for this framework, but they showed worse results than those obtained using LDAs in our experiments.

3.3. Quasi-Restricted Equivalent Functions and quasi-Restricted Dissimilarity Functions

In this section we present the concept of Q-REF and Q-RDF. We recall here the notions of Restricted Equivalent Functions (REFs) and Restrict Dissimilarity Functions (RDFs) [31, 32] that will be the basis for Q-REF and Q-RDF.

Definition 2. *A function $c : [0,1] \to [0,1]$ is called a strong negation if and only if there exists an automorphism ϕ such that $c(x) = \phi^{-1}(1 - \phi(x))$.*

Definition 3. *A function $s : [0,1]^2 \to [0,1]$ is called a REF if:*

1. *$s(x,y) = s(y,x)$;*
2. *$s(x,y) = 1$ if and only if $x = y$;*
3. *$s(x,y) = 0$ if and only if $\{x,y\} = \{0,1\}$;*
4. *$s(x,y) = s(c(x), c(y))$ for all $x, y \in [0,1], c$ being a strong negation.*
5. *If $x \leq y \leq z$ then $s(x,z) \leq s(x,y)$ and $s(x,z) \leq s(y,z)$.*

Definition 4. *A function $d : [0,1]^2 \to [0,1]$ is called a RDF if:*

1. *$d(x,y) = d(y,x)$;*
2. *$d(x,y) = 0$ if and only if $x = y$;*
3. *$d(x,y) = 1$ if and only if $\{x,y\} = \{0,1\}$;*
4. *If $x \leq y \leq z$ then $d(x,y) \leq d(x,z)$ and $d(y,z) \leq d(x,z)$.*

In order to deal with more than two inputs, properties are relaxed to introduce the notions of Q-REF function and Q-RDF.

Definition 5. *Let $n \geq 1$. A Q-REF function is a function $H_s : [0,1]^{n+1} \to [0,1]$ such that:*

$$H_s(X,y) = H_s(x_1, \dots, x_n, y) = 1 \text{ if } x_1 = \dots = x_n = y. \qquad (1)$$

Note that REFs are specific instances of Q-REF functions. And analogously:

Definition 6. *Let $n \geq 1$. A Q-RDF function is a function $H_d : [0,1]^{n+1} \to [0,1]$ such that:*

$$H_d(X,y) = H_d(x_1, \dots, x_n, y) = 0 \text{ if } x_1 = \dots = x_n = y. \qquad (2)$$

Again, RDFs are specific instances of Q-RDF functions. First of all, observe that these two types of functions are closely related. In fact, we have the following straightforward result.

Proposition 7. *Let $n : [0,1] \to [0,1]$ be a decreasing function such that $n(0) = 1$ and $n(1) = 0$ (a negation). Then, a function $H_s : [0,1]^n \to [0,1]$ is a Q-REF function if and only if $n(H_s)$ is a Q-RDF function.*

We can build general Q-REF and Q-REF functions as follows.

Proposition 8. *Let $h_{s1}, \dots, h_{sn} : [0,1]^2 \to [0,1]$ be a family of Q-REF functions and let $A : [0,1]^n \to [0,1]$ be an aggregation function. Then, $H_s^A(x_1, \dots, x_n, y) = A(h_{s1}(x_1, y), \dots, h_{sn}(x_n, y))$ is also a Q-REF function.*

Proposition 9. *Let $h_{d1}, \dots, h_{dn} : [0,1]^2 \to [0,1]$ be a family of Q-RDF functions and let $A : [0,1]^n \to [0,1]$ be an aggregation function. Then, $H_d^A(X, y) = A(h_{d1}(x_1, y), \dots, h_{dn}(x_n, y))$ is also a Q-RDF function.*

Proposition 10. *Let $H_{s1}, H_{s2} : [0,1]^n \to [0,1]$ be two Q-REF functions. Then, for every $\alpha \in [0,1]$*

$$\alpha H_{s1} + (1 - \alpha) H_{s2} \tag{3}$$

is also a Q-REF function.

Proposition 11. *Let $H_{d1}, H_{d2} : [0,1]^n \to [0,1]$ be two Q-RDF functions. Then, for every $\alpha \in [0,1]$*

$$\alpha H_{d1} + (1 - \alpha) H_{d2} \tag{4}$$

is also a Q-RDF function.

Now we consider the convex combination of a Q-REF and a Q-RDF function. If $x_1 = \cdots = x_n = y$, we have that:

$$\alpha H_d(x_1, \dots, x_n, y) + (1 - \alpha) H_s(x_1, \dots, x_n, y) = 1 - \alpha \tag{5}$$

So:

Proposition 12. *Let $H_d, H_s : [0,1]^n \to [0,1]$ be a Q-REF and a Q-RDF function, respectively. Then, for any $\alpha \in [0,1[$, the function:*

$$H(X, y) = \min\left(\frac{\alpha H_d(X, y) + (1 - \alpha) H_s(X, y)}{1 - \alpha}, 1\right) \tag{6}$$

is a Q-REF function.

3.4. Multi-cost aggregation-choosing functions

A penalty function is characterized using a disagreement measure that quantifies how different the inputs X are with respect to the resulting aggregated value y. The use of penalty functions mitigates the problem of choosing an appropriate aggregation function: given a disagreement measure, the one whose output minimizes the disagreement measure will be chosen. The most common disagreement measure is the quadratic error, however, the arithmetic mean will always deliver the best result according to this measure [16].

To solve this problem we propose the MCAs, that present two novelties compared to the already existing penalty-based aggregation functions:

- In order to measure the disagreement, we consider a cost function. We do so as a cost function can be applied in situations where the term consensus would not be adequate. For example, in a N-class classification problem, a result of 1/N probability for a specific class, indicates that the output does not contain almost any information. In this case, the cost function can be used to penalize this kind of outcome. In this manuscript, we chose Q-REFs and Q-RDFs as cost functions, as studied in Section 3.3.
- The use of a convex combination of two functions instead of a single function avoids trivial results such as the one regarding the quadratic cost, which will always be minimized by the arithmetic mean independently of the input data.

A schematic view of the aggregation process using a MCA can be found in Fig. 3.

3.4.1. Costs used

We have considered a set of Q-RDFs and Q-REF measures as cost functions. As studied in Section 3.3, and depending on the mixed functions, their convex combination is also a Q-REF or a Q-RDF. Given a vector X of size n, where each element of X is contained in the unit interval the Q-RDFs measures studied are the following:

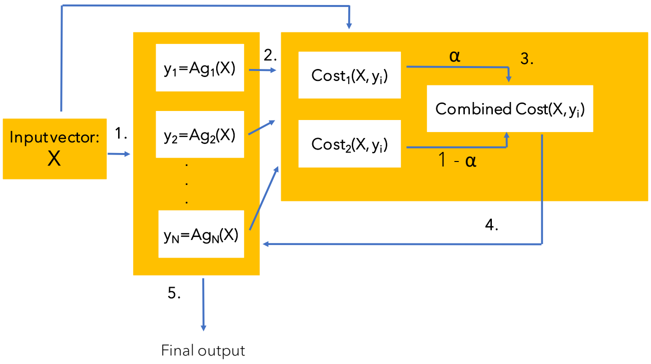


Figure 3: Visual scheme for the MCA aggregation process. (In the case of the BCI framework in Fig. 1, X is the output of the LDA classifiers). **1**. We compute all the possible aggregations. **2**. We compute both cost functions for each aggregation output (y_i) with respect to the input vector. **3**. We combine both costs for each aggregation with the mixing parameter α. **4**. We select the aggregation with the least cost value. **5**. That aggregation is the final output of the MCA.

- Huber loss:

$$h(x_i,y) = \begin{cases} (x_i-y)^2 & (x_i-y)^2 \leq M \\ 2*M*(x_i-y)^2 - M*M & (x_i-y)^2 \geq M \end{cases} \tag{7}$$

  where $H(X,y) = \frac{1}{n}\sum_{i=1}^{n} h(x_i,y)$.

  (We use $M = 0.3$ for our experimentation)

- Quadratic cost:

$$H(X,y) = \frac{1}{n}\sum_{i=1}^{n}(x_i-y)^2 \tag{8}$$

- Optimistic cost:

$$H(X,y) = (max(X)-y)^2 \tag{9}$$

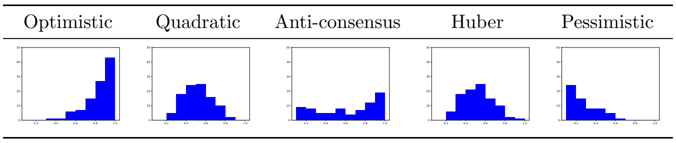


Figure 4: Histogram of aggregated values for the optimistic, the quadratic, anti-consensus, Huber and pessimistic costs using the maximum, minimum, arithmetic mean and median as possible aggregations in the MCA, for a random sample of 100 vectors of size 5 in the $[0, 1]$ range. We represent in the x axis the α value and, in the y axis, the frequency of the aggregated output values in each range for each sampled random vector.

- Pessimistic cost:

$$H(X, y) = (min(X) - y)^2 \tag{10}$$

The Q-REF measure studied is:

- Anti-consensus cost:

$$H(X, y) = \frac{1}{n}\sum_{i=1}^{n}(1 - (x_i - y)^2) \tag{11}$$

Fig. 4 shows the effects of a penalty aggregation using the classical aggregations applied to the BCI data, with a sample of 100 five dimensional (5-D) random vectors with numbers in [0,1]. The histograms are computed over the results of aggregating 100 5-D random vectors. It is visible that the optimistic and pessimistic costs have a "skewing effect", so that the histogram is sharply moved to greater and lower values, respectively. It can also be observed that there are two very similar cost functions: the quadratic and the Huber costs. This is expected as they only differ in "extreme" values. Finally, the anti-consensus cost exhibits the most disperse histogram.

3.4.2. Combining costs

The combination of two costs using a convex combination requires an $\alpha \in]0, 1[$ parameter. Depending which are the functions to be combined, Q-REFs or Q-RDFs, different formulas should be used:

- Both are the same type:

$$\text{Combined Cost} = \alpha * cost_1 + (1 - \alpha) * cost_2 \tag{12}$$

- One is a Q-REF and the other is a quasi-dissimilarity:

$$\text{Combined cost} = \min(\frac{\alpha cost_1 + (1 - \alpha) cost_2}{1 - \alpha}, 1) \tag{13}$$

Thus, the combined cost will be another Q-RDF when both cost_1 and cost_2 are both Q-RDF, and a Q-REF otherwise.

Fig. 5 shows how the cost functions behave for a five-dimensional vector of random numbers in the interval $[0, 1]$: $(0.60, 0.85, 0.61, 0.52, 0.52)$.

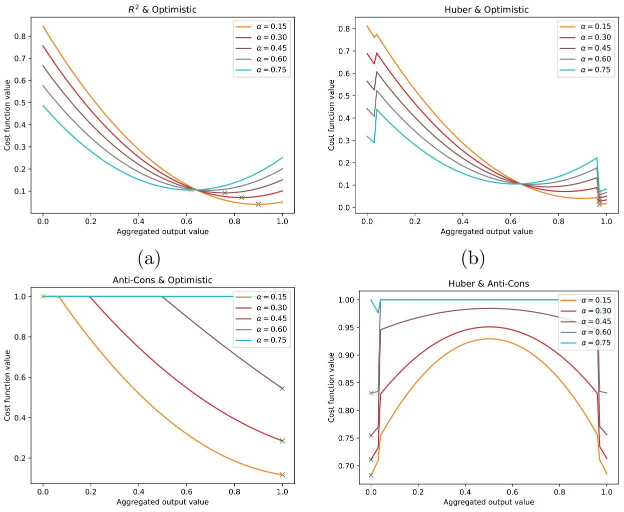
In order to show how each cost combination works, we computed each of them varying the parameter α within the $]0, 1[$ interval. We also marked the preferred value for each one. Fig. 5a and Fig. 5b correspond to Q-RDFs and the Fig. 5c and Fig. 5d are Q-REFs.

Fig. 6 studies the effect of different α values in the quadratic and optimistic cost based on the same random vectors as before and shows that indeed the α parameter has a notorious influence in the chosen aggregation.

3.5. Selecting the α parameter in a Multi-cost aggregation-choosing function for a supervised classification task

As studied in Section 3.4, α plays a crucial role in the output of a MCA. Choosing the optimal value for this parameter is not an easy task because it heavily depends on the application.

Although some fixed α value might work sufficiently well for some applications, the fine-tuning of this parameter can also increase the performance of



(a)

(b)

(c)

(d)

Figure 5: Effect of different α parameters for a vector of five, randomly chosen numbers $\in [0, 1]$: $(0.60, 0.85, 0.61, 0.52, 0.52)$. The × marks the minimum for each α parameter in each error configuration.

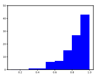
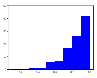
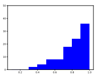
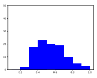
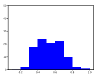
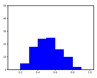
$\alpha = 0.0$	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$	$\alpha = 1.0$
					

Figure 6: Histogram of aggregated values for the quadratic & optimistic cost using the maximum, minimum, arithmetic mean and median as possible aggregations in the MCA, for a random sample of 100 vectors of size 5 in the $[0, 1]$ range, using different α values. We represent in the x axis the α value and, in the y axis, the frequency of the aggregated output values in each range for each sampled random vector.

supervised classification scenarios. As α is restricted to the $]0, 1[$ interval, a dense sampling Montecarlo optimization with accuracy as target metric is a good option to select α. Nevertheless, when a system is composed of more than one aggregation process and more than one MCA, the optimization needs to be performed over a vector of numbers instead of just one value. Depending on the size of the vector, it is still possible to optimize it performing the same procedure as for a single value. However, one of the key ideas of the original penalty-based aggregations is to find a suitable aggregation for each vector of inputs. By choosing the same α parameter for each individual vector of inputs, this philosophy is somewhat disregarded. In that case, computing an adaptive α, chosen according to the vector of inputs appears more appropriate. We named this procedure the adaptive MCA.

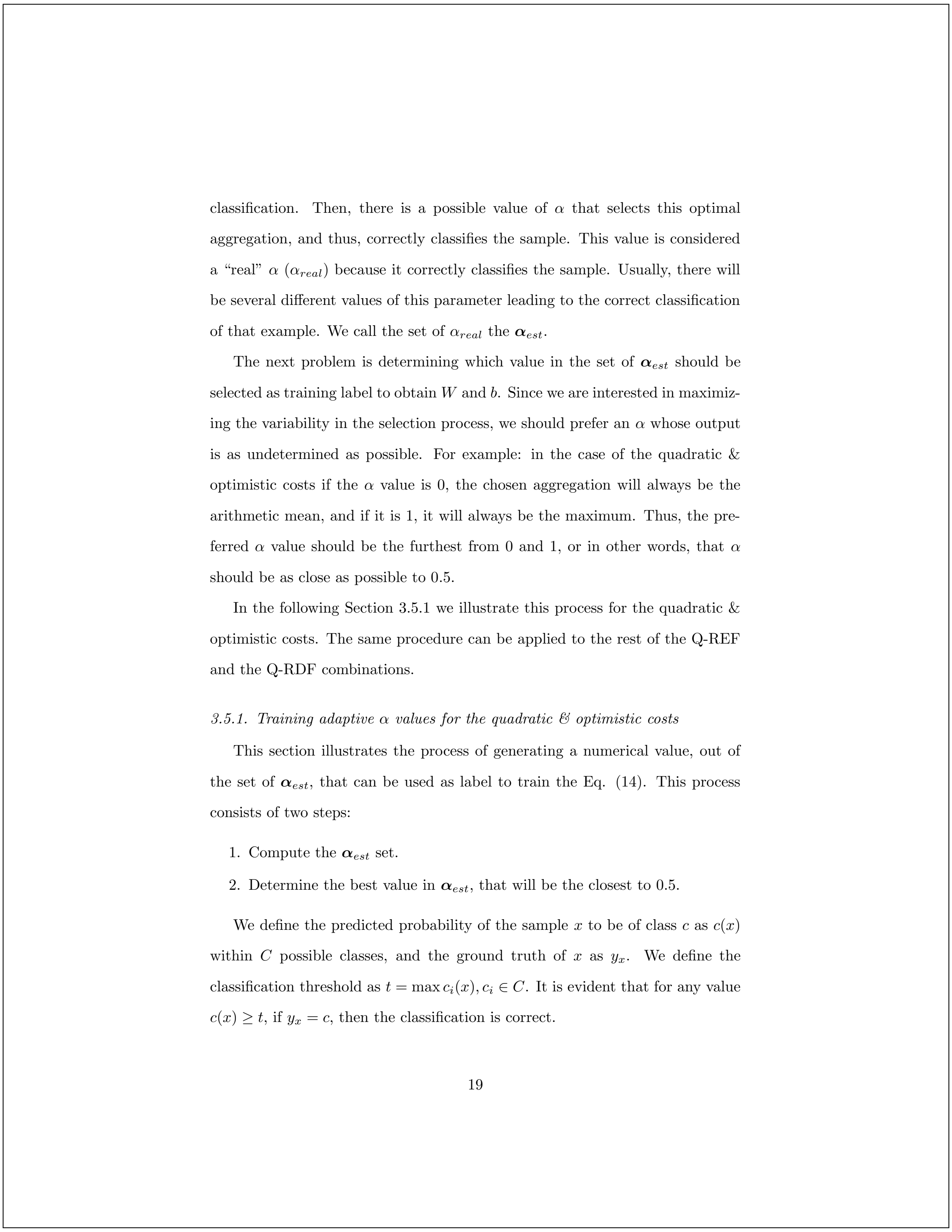
The adaptive MCA computation carries an additional difficulty, as we need to somehow relate our input vector with the final outcome, which is the label for each sample. We propose the use of a regression:

$$\alpha = f(WX + b) \tag{14}$$

where f is an activation function, W is the weight matrix, X is the input vector and b is the bias.

In this formula, both W and b matrices should be optimized. In this case a Montecarlo optimization using the accuracy as the target metric is not appropriate because the size of X might be too large, turning the optimization unstable due to the "curse of dimensionality". Thus, we propose to learn W and b using a gradient descent optimization, where a set of initial "real" α values is necessary.

Although "real" α values do not exist, there is a ground truth label for each example. Suppose an aggregation function exists whose result leads to a correct



classification. Then, there is a possible value of α that selects this optimal aggregation, and thus, correctly classifies the sample. This value is considered a "real" α (α_{real}) because it correctly classifies the sample. Usually, there will be several different values of this parameter leading to the correct classification of that example. We call the set of α_{real} the $\boldsymbol{\alpha}_{est}$.

The next problem is determining which value in the set of $\boldsymbol{\alpha}_{est}$ should be selected as training label to obtain W and b. Since we are interested in maximizing the variability in the selection process, we should prefer an α whose output is as undetermined as possible. For example: in the case of the quadratic & optimistic costs if the α value is 0, the chosen aggregation will always be the arithmetic mean, and if it is 1, it will always be the maximum. Thus, the preferred α value should be the furthest from 0 and 1, or in other words, that α should be as close as possible to 0.5.

In the following Section 3.5.1 we illustrate this process for the quadratic & optimistic costs. The same procedure can be applied to the rest of the Q-REF and the Q-RDF combinations.

3.5.1. Training adaptive α values for the quadratic & optimistic costs

This section illustrates the process of generating a numerical value, out of the set of $\boldsymbol{\alpha}_{est}$, that can be used as label to train the Eq. (14). This process consists of two steps:

1. Compute the $\boldsymbol{\alpha}_{est}$ set.
2. Determine the best value in $\boldsymbol{\alpha}_{est}$, that will be the closest to 0.5.

We define the predicted probability of the sample x to be of class c as $c(x)$ within C possible classes, and the ground truth of x as y_x. We define the classification threshold as $t = \max c_i(x), c_i \in C$. It is evident that for any value $c(x) \geq t$, if $y_x = c$, then the classification is correct.

As aforementioned the quadratic error favours the arithmetic mean over the rest of the aggregations, and the optimistic error, favours the maximum. We consider the convex combination of both errors and the mixing parameter α:

$$Cost(X)_\alpha = \alpha * mean(X) + (1 - \alpha) * max(X) \tag{15}$$

The MCA that uses this cost increases with respect to the α value, because as the value of α grows, the preferred value in the error formula gets closer to the maximum.

Supposing that an α’ exists such that for the class c, and $c = y_x$, the MCA$_{\alpha'}(x) = t$, all $\alpha'' > \alpha'$ will result in a MCA$_{\alpha''}(x) >= MCA_{\alpha'}(x) >= t$, which will result in correct classification. This means that $\boldsymbol{\alpha}_{est}$ are all the α values bigger than α'. Then, the optimal $\boldsymbol{\alpha}_{est}$ is just the closest to 0.5.

The process is very similar for any other combination of Q-REF and Q-RDF functions, but if the combined cost is not monotone with respect to α, then $\boldsymbol{\alpha}_{est}$ can be disjoint.

Example 3.1. *Taking a vector of five random numbers:* $\mathbf{x} = [0.4, 0.9, 0.1, 0.5, 0.3]$, *we consider these five random numbers the output of five classifiers, i.e. the probability of a sample to be of class y, being y the real label of that sample.*

We select a MCA that uses the maximum and mean cost, and chooses among the average (0.44)*, median* (0.40)*, minimum* (0.1) *and maximum* (0.90) *aggregations. Then, for any $\alpha < 0.5$, the MCA will choose the average, and for any $\alpha > 0.5$ the MCA will select the maximum. For $\alpha = 0.5$ both values are eligible. Since the average is 0.44, if we aggregate using this value, the final result will not correctly classify* $\mathbf{x}$. *If we aggregate using the maximum, then the aggregation will correctly classify the sample. So, α_{est} in this case will be all α values greater than* 0.5.

As final training label we take the immediate value following 0.5 *and adjust it to the desired precision. For example, if we consider decimals until the third digit, the target α_{real} to learn for* $\mathbf{x}$ *would be* 0.501.

3.6. Multi-cost aggregation-choosing functions in the Brain Computer Interface Framework

We use the MCA functions in the aggregation function in the decision making phase of the BCI framework. Each MCA is composed of a set of possible aggregations to choose from and a cost function. In the case of the adaptive-MCA, it is also composed of a weight matrix and a bias vector. We used a set composed of four classical aggregations: minimum, maximum, median, and the arithmetic mean. We tested all possible combinations of Q-REFs and Q-RDFs and presented them in Section 3.4.1.

In the case of using a non adaptive MCA, the mixing parameter was learnt using a Montecarlo sampling of 200 possible α values in the $]0,1[$ range. On the other hand, recall that in the case of the adaptive MCA we need to establish the X matrix and the activation function f of Eq. (14) to apply the procedure detailed in Section 3.5. Matrix X corresponds to the outputs of all the classifiers in the BCI framework for each sample, whereas f is a linear activation function ($f(x) = x$). Then, W and b in Eq. (14) are learnt using gradient descent optimization.

4. Results

In this section we discuss the outcomes of applying our new approaches to the BCI competition IV dataset 2a (IV-2a) [18] and the Clinical BCI Challenge WCCI 2020 dataset (CBCIC) [19].

- **Dataset 1**: The IV-2a dataset has been extensively used to test different BCI systems (see for example, [33, 34, 35]). It consists of four motor imagery tasks (tongue, foot, left-hand and right-hand) performed by 9 volunteers. For each task, 22 EEG channels were collected, with a total of 288 trials for each participant. Trials are evenly distributed among the 4 classes. For this dataset, we studied the classification task from two different perspectives: binary classification of the left and right hand classes, which is a common choice of tasks in the literature [35, 36]; and four-class classification: left hand, right hand, foot and tongue.

- **Dataset 2**: the CBCIC dataset consist of brain imaging signals from 10 hemiparetic stroke patients with hand functional disability in a rehabilitation task. The data contains 80 diferent trials of left/right hand movements. Decoding motor cortical signals of brain-injured presents several challenges as the presence of irregular because of the altered neurodynamics [19].

For both datasets, the evaluation process is the same. Each participant's dataset was randomly sampled in ten different partitions (each with 50% train and 50% test trials). A total of 90, respectively 80 datasets were generated for the IV-a Competition and CBCIC datasets. The final performance of each configuration was obtained averaging each single dataset accuracy. The results were obtained using different aggregation functions in the decision making phase and compared the newly proposed MCAs. Both the adaptive and the non-adaptive mixing parameter were employed with a set of standard aggregations and also with the already existing penalty-based aggregation functions.

Furthermore, results for each individual subject are available in the following GitHub repository: `https://github.com/Fuminides/MCA_BCI_results`.

4.1. Results for left/right hand motor imagery classification with stroke patients (CBCIC) and BCI competition IV-2a datasets

Table 1 shows the results for the binary classification using the state-of-art BCI framework with the arithmetic mean as the fusion function for the classifiers output. Recall here that the choice of the penalty-based aggregation function

Table 1: Results using Penalty-based aggregation/arithmetic mean.

Dataset	Aggregation	Accuracy
IV-2a dataset	Average/Classic Penalty-based aggregation	0.7974
CBCIC dataset	Average/Classic Penalty-based aggregation	0.8215

Table 2: Accuracy results for binary classification using MCAs optimized with Montecarlo Sampling in the binary task.

Dataset		Quadratic	Optimistic	Huber	Pessimistic
IV-2a	Optimistic	0.7904			
	Huber	0.7960	0.7931		
	Pessimistic	0.7933	0.7938	0.7915	
	Anti-consensus	0.7955	0.8022	0.7939	**0.8030**
CBCIC	Optimistic	0.8123			
	Huber	0.8215	0.8142		
	Pessimistic	0.8113	0.8000	0.8224	
	Anti-consensus	0.8215	0.8221	**0.8231**	0.8215

is always the arithmetic mean, thus the results are the same for both.

Table 2 displays the results for all the possible MCA functions. The selected aggregation functions are a set of classical aggregation procedures: arithmetic mean, median, minimum and maximum. In this case, α was found with a simple Montecarlo sampling optimization, using a ten-fold validation on the train set to determine its performance. Table 3 presents analogous results to those in Table 2, but using the algorithm proposed in Section 3.5.1 to learn the α parameter.

These tables show that the best result is obtained for a MCA with α set by the procedure described in Section 3.5, resulting in 0.8038 of accuracy in the IV-2a dataset, and 0.8231 in the CBCIC dataset. The second best result is obtained for a MCA with a Montecarlo optimization. Both MCA optimization algorithms improve the result of the classical arithmetical mean: 0.7974 and 0.8224 for the IV-2a and CBCIC dataset, respectively. We performed a Friedman test, as both populations were not normal according to Shapiro-Wilk test. However, no statistical differences were found.

Table 3: Accuracy results for the adaptive MCA optimized with the algorithm in Section 3.5 in the binary task.

Dataset		Quadratic	Optimistic	Huber	Pessimistic
IV-2a	Optimistic	0.8000			
	Huber	0.7974	0.7994		
	Pessimistic	0.8000	0.7798	0.7994	
	Anti-consensus	0.7974	**0.8038**	0.7970	**0.8038**
CBCIC	Optimistic	0.8123			
	Huber	0.8215	0.8215		
	Pessimistic	0.8113	0.8132	**0.8224**	
	Anti-consensus	0.8215	0.8212	0.8221	0.8136

4.2. Results for 4-class motor-imagery classification problem (BCI competition IV-2a)

The 4- class problem is analogous to the left/right hand problem including "foot" and "tongue" tasks, which are noticeably harder to discriminate [18]. To study this problem we have performed similar experiments to those of the left/right hand classification task.

Table 4 shows the results for the state-of-art BCI framework using the arithmetic mean, which is similar to computing the classical penalty-based aggregation. The obtained accuracy was 0.6056.

Table 5 displays the results for the state-of-art BCI aggregation framework using the MCA functions where the α parameter was optimized with the Montecarlo sampling algorithm. We found many combinations of costs that resulted in MCAs surpassing the result of the arithmetic mean. The best result found here was 0.6243 of accuracy.

Finally, Table 6 presents the results for the traditional BCI framework using the MCA functions where the α parameter was optimized with the algorithm in Section 3.5. Here, the best result was 0.6167, which was again better than the one obtained using the arithmetic mean, but worse than the one using the Montecarlo sampling optimization.

Table 4: Accuracy 4-class classification results using Penalty-based aggregation/arithmetic mean in the IV-2a dataset.

Dataset	Aggregation	Accuracy
IV-2a	Average/Classic Penalty-based aggregation	0.6056

Table 5: Accuracy 4-class classification results using MCAs optimized with Montecarlo Sampling.

Dataset		Quadratic	Optimistic	Huber	Pessimistic
IV-2a	Optimistic	0.6066			
	Huber	0.6041	0.6040		
	Pessimistic	0.6027	0.6046	0.5993	
	Anti-consensus	0.6056	0.6087	0.6018	**0.6243**

According to a Shapiro-Wilk test, the accuracy populations were not normal. So, we used a Friedman test followed by pairwise comparisons with Wilcoxon post-hoc tests to look for statistical differences. The resulting P-values are reported in Table 7. We found the Montecarlo optimization to significantly outperform the rest.

5. Comparison with other Motor Imagery-Brain Computer Interface decoding methods

In this Section we compare our results with two other MI-BCI systems. We employed both the IV-2a and the CBCIC datasets. The selected BCI frameworks are described in the following:

1. Multimodal Fuzzy Fusion framework (MFF) [12]: in this work the authors

Table 6: Accuracy 4-class classification results adaptive MCAs optimized with the algorithm in Section 3.5

Dataset		Quadratic	Optimistic	Huber	Pessimistic
IV-2a	Optimistic	0.6124			
	Huber	0.6056	0.6113		
	Pessimistic	0.6050	0.5966	0.5990	
	Anti-consensus	0.6056	0.6148	0.6033	**0.6167**

Table 7: Statistical significances in the four classes classification problem with Wilcoxon post-hoc for the different MPA approaches and the arithmetic mean.

Dataset		Arithmetic mean	MCA Montecarlo
IV-2a	MCA Montecarlo	$P < .001$	
	MCA adaptive	$P < .001$	$P < .001$

use a Fast Fourier transform to extract features from the original EEG data, then they construct a classifier ensemble using different types of classifiers and a fuzzy integral.

2. One Versus One and Gradient Boosting [17]: the authors used Gradient Boosting classifiers [37] to select the optimal classification features. They structured the decision making phase with different One versus One (OVO) strategies: a classical OVO, and a tree structure for the OVO classifiers (tree-OVO).
3. Multiscale CSP [38]: the authors extended CSP using different time windows, to obtain features from different temporal scales, which then are used to train a SVM classifier.
4. EEG net [39]: in this work, the authors proposed a specific architecture of a a Convolutional Neural Network for EEG signals, in order to incorporate in the network different well-known concepts of feature extraction in BCI.
5. Shallow and Deep nets [40]: are two convolutional neural networks, composed of 2 and 4 blocks of convolution and max pooling blocks.

In order to compare our feature extraction method with others, we also used the feature extraction method developed in [38] with the proposed MCA. In this framework, the features from different time windows are concatenated and fed to a classifier. In order to use the proposed MCA, instead of concatenating these features into a single vector, we form k different vectors concatenating the features from adjacent frequencies. For each of these feature vectors we train a classifier, and then we fuse the logits from these classifiers using a MCA. We call this framework the multiscale MCA

We performed these comparisons using the same procedure as in Section 4 and that we summarize here: we randomly sampled 10 partitions composed of 50% train and 50% test data for each subject. This resulted in 90 different datasets for the IV-2a competition data, and 80 datasets for the CBCIC. As evaluation metric, we used the mean accuracy obtained in the test partitions.

Table 8 shows the results for each of the different configurations tested for the IV-2a dataset. Table 9 shows the same comparison for the CBCIC dataset. We found that the our method performed best for the CBCIC dataset, and that the Multiscale MCA over performed the rest for the IV-2a dataset. In this configuration of the Multiscale MCA we used two feature vectors and the Huber & Anti-consensus cost.

Table 10 shows the results for the Wilcoxon post-hoc after Friedman test, comparing the MCA Montecarlo with the rest of the frameworks tested for the IV-2a dataset. We found that MCA Montecarlo significantly outperforms OVO, tree-OVO frameworks but the MFF performed statistically better than our proposal. Table 11 shows the analogous results for the CBCIC dataset. In this case we found that our method performed significantly better than the MFF.

6. Conclusions and Future work

In this paper we introduced the combination of two generalized versions of REFs and RDFs cost functions to choose an optimal aggregation regarding a vector of inputs. We showed that this technique is able to enhance the classifier fusion phase in two BCI frameworks and can improve the results of the arithmetic mean (and subsequently, the classical penalty-based aggregations) for both binary and multiclass MI classification problems of the BCI Competition

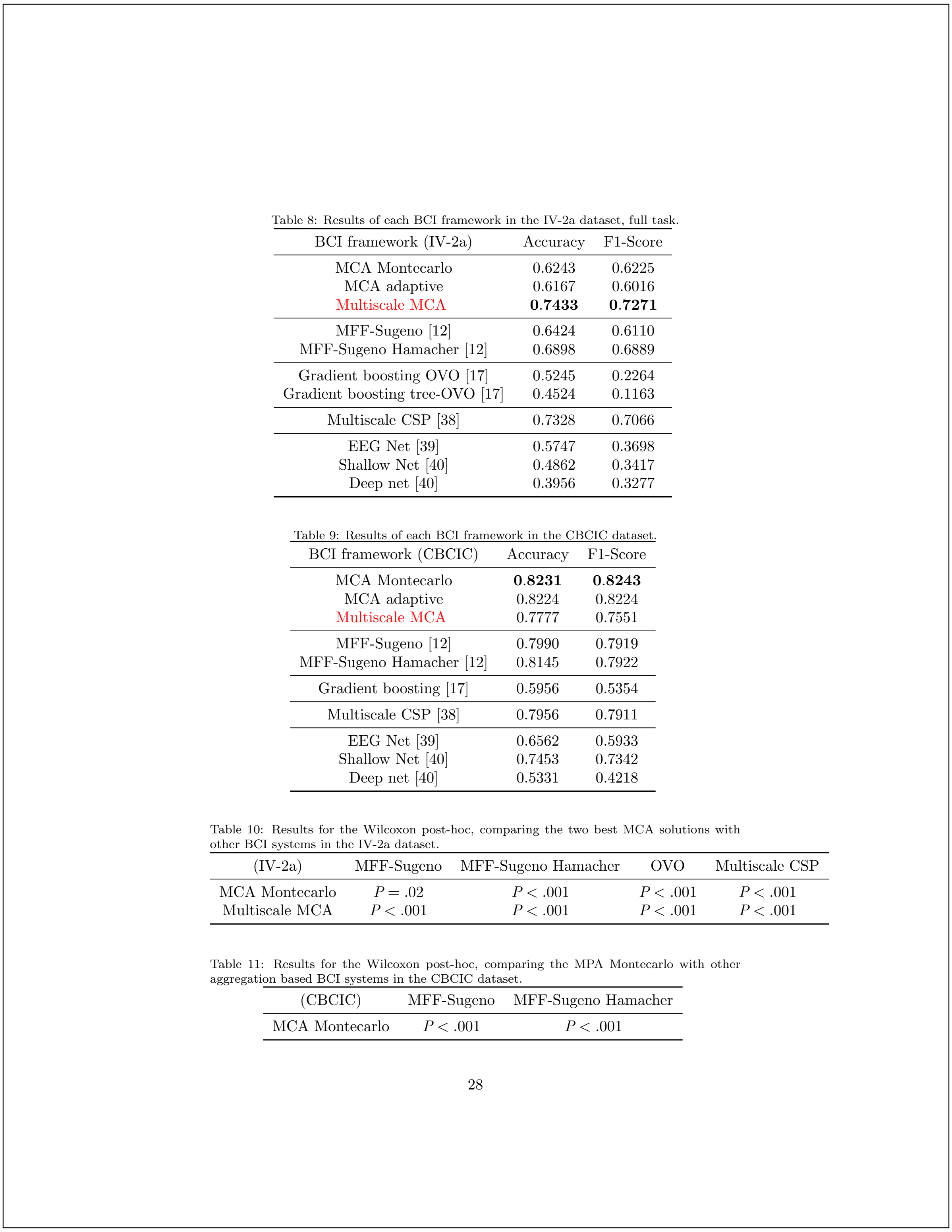


Table 8: Results of each BCI framework in the IV-2a dataset, full task.

BCI framework (IV-2a)	Accuracy	F1-Score
MCA Montecarlo	0.6243	0.6225
MCA adaptive	0.6167	0.6016
Multiscale MCA	**0.7433**	**0.7271**
MFF-Sugeno [12]	0.6424	0.6110
MFF-Sugeno Hamacher [12]	0.6898	0.6889
Gradient boosting OVO [17]	0.5245	0.2264
Gradient boosting tree-OVO [17]	0.4524	0.1163
Multiscale CSP [38]	0.7328	0.7066
EEG Net [39]	0.5747	0.3698
Shallow Net [40]	0.4862	0.3417
Deep net [40]	0.3956	0.3277

Table 9: Results of each BCI framework in the CBCIC dataset.

BCI framework (CBCIC)	Accuracy	F1-Score
MCA Montecarlo	**0.8231**	**0.8243**
MCA adaptive	0.8224	0.8224
Multiscale MCA	0.7777	0.7551
MFF-Sugeno [12]	0.7990	0.7919
MFF-Sugeno Hamacher [12]	0.8145	0.7922
Gradient boosting [17]	0.5956	0.5354
Multiscale CSP [38]	0.7956	0.7911
EEG Net [39]	0.6562	0.5933
Shallow Net [40]	0.7453	0.7342
Deep net [40]	0.5331	0.4218

Table 10: Results for the Wilcoxon post-hoc, comparing the two best MCA solutions with other BCI systems in the IV-2a dataset.

(IV-2a)	MFF-Sugeno	MFF-Sugeno Hamacher	OVO	Multiscale CSP
MCA Montecarlo	$P = .02$	$P < .001$	$P < .001$	$P < .001$
Multiscale MCA	$P < .001$	$P < .001$	$P < .001$	$P < .001$

Table 11: Results for the Wilcoxon post-hoc, comparing the MPA Montecarlo with other aggregation based BCI systems in the CBCIC dataset.

(CBCIC)	MFF-Sugeno	MFF-Sugeno Hamacher
MCA Montecarlo	$P < .001$	$P < .001$

IV 2a and CBCIC datasets.

For the latter dataset, our BCI framework performed better than the Deep Learning, OVO, Multimodal Fusion and Multiscale CSP proposals regardless the aggregation function chosen. We also found that the best MCAs computed included the Anti-Consensus cost, which favors values that differ from the consensus. This result suggests that the most useful aggregated values to perform classification can be different to the original consensus of the classifiers. This idea differs from the original penalty functions intention, which was to measure disagreement in order to choose the value that minimizes it.

Future research shall aim at improving the accuracy of the system by studying different ways to learn which costs should be combined for a given task. We also intend to study the trade-off between diversity and accuracy in the classifiers to aggregate, as the more diverse these outputs are, the more meaningful the aggregation process can be.

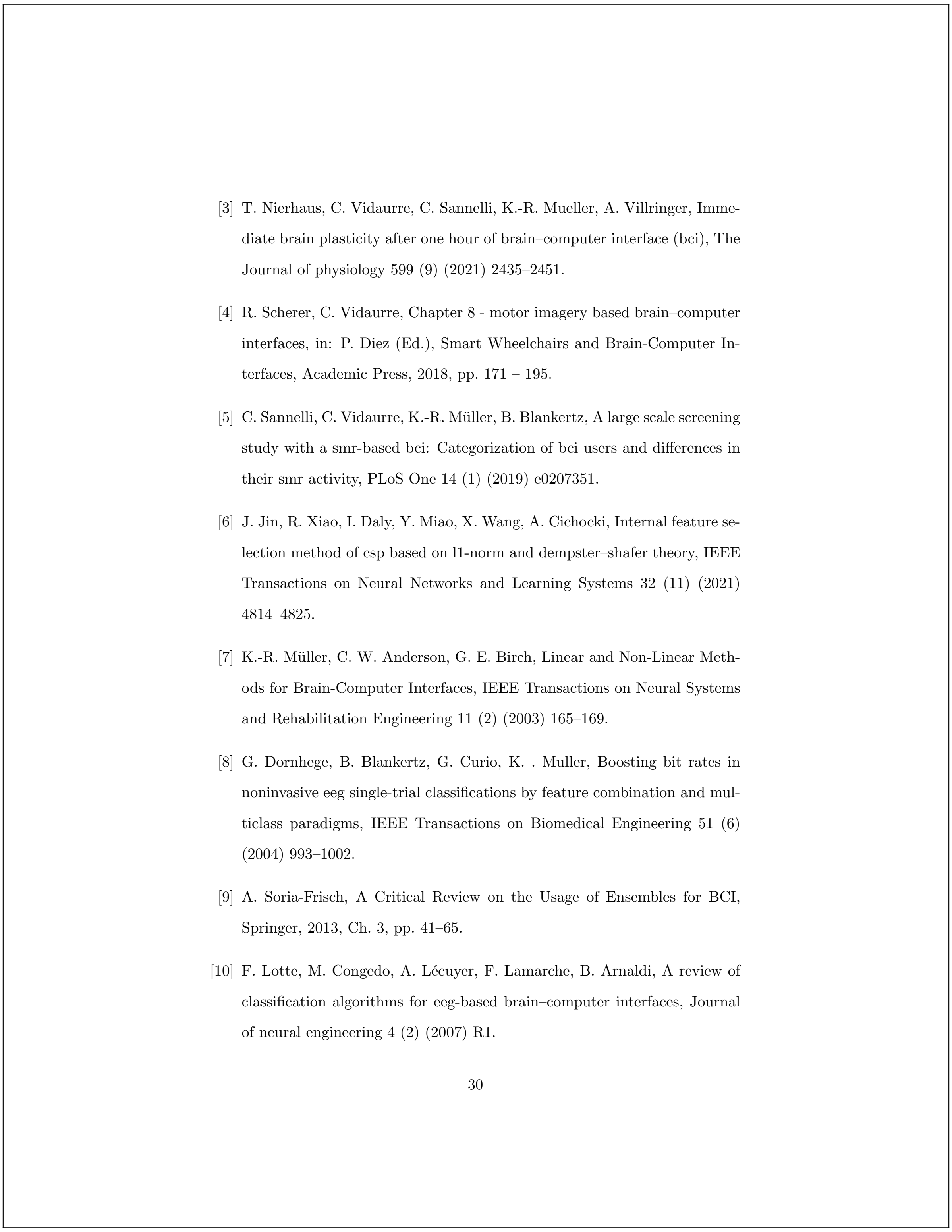
7. Acknowledgments

Javier Fumanal Idocin, Javier Fernandez, and Humberto Bustince's research has been supported by the project PID2019-108392GB I00 (AEI/10.13039/501100011033). Carmen Vidaurre research has been funded by the project RyC-2014-15671.

References

[1] Y.-K. Wang, T.-P. Jung, C.-T. Lin, Eeg-based attention tracking during distracted driving, IEEE transactions on neural systems and rehabilitation engineering 23 (6) (2015) 1085–1094.

[2] M.-H. Lee, S. Fazli, J. Mehnert, S.-W. Lee, Subject-dependent classification for robust idle state detection using multi-modal neuroimaging and data-fusion techniques in bci, Pattern Recognition 48 (8) (2015) 2725–2737.



[3] T. Nierhaus, C. Vidaurre, C. Sannelli, K.-R. Mueller, A. Villringer, Immediate brain plasticity after one hour of brain–computer interface (bci), The Journal of physiology 599 (9) (2021) 2435–2451.

[4] R. Scherer, C. Vidaurre, Chapter 8 - motor imagery based brain–computer interfaces, in: P. Diez (Ed.), Smart Wheelchairs and Brain-Computer Interfaces, Academic Press, 2018, pp. 171 – 195.

[5] C. Sannelli, C. Vidaurre, K.-R. Müller, B. Blankertz, A large scale screening study with a smr-based bci: Categorization of bci users and differences in their smr activity, PLoS One 14 (1) (2019) e0207351.

[6] J. Jin, R. Xiao, I. Daly, Y. Miao, X. Wang, A. Cichocki, Internal feature selection method of csp based on l1-norm and dempster–shafer theory, IEEE Transactions on Neural Networks and Learning Systems 32 (11) (2021) 4814–4825.

[7] K.-R. Müller, C. W. Anderson, G. E. Birch, Linear and Non-Linear Methods for Brain-Computer Interfaces, IEEE Transactions on Neural Systems and Rehabilitation Engineering 11 (2) (2003) 165–169.

[8] G. Dornhege, B. Blankertz, G. Curio, K. . Muller, Boosting bit rates in noninvasive eeg single-trial classifications by feature combination and multiclass paradigms, IEEE Transactions on Biomedical Engineering 51 (6) (2004) 993–1002.

[9] A. Soria-Frisch, A Critical Review on the Usage of Ensembles for BCI, Springer, 2013, Ch. 3, pp. 41–65.

[10] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, B. Arnaldi, A review of classification algorithms for eeg-based brain–computer interfaces, Journal of neural engineering 4 (2) (2007) R1.

[11] D. Achanccaray, K. Acuna, E. Carranza, J. Andreu-Perez, A virtual reality and brain computer interface system for upper limb rehabilitation of post stroke patients, in: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2017, pp. 1–5.

[12] L.-W. Ko, Y.-C. Lu, H. Bustince, Y.-C. Chang, Y. Chang, J. Ferandez, Y.-K. Wang, J. A. Sanz, G. P. Dimuro, C.-T. Lin, Multimodal fuzzy fusion for enhancing the motor-imagery-based brain computer interface, IEEE Computational Intelligence Magazine 14 (1) (2019) 96–106.

[13] G. Beliakov, H. Bustince, T. C. Sánchez, A practical guide to averaging functions, Vol. 329, Springer, 2016.

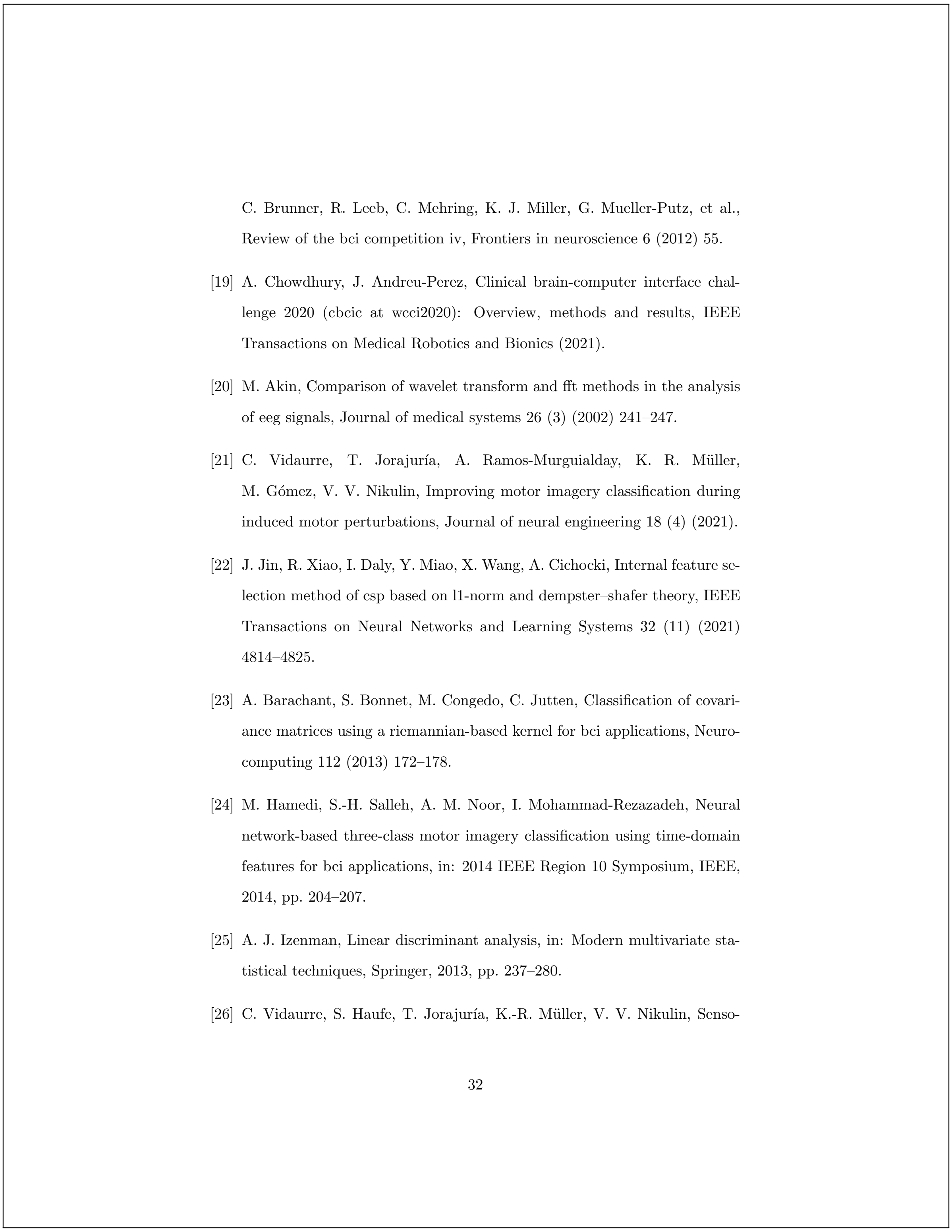
[14] J. Fumanal-Idocin, Z. Takac, J. Fernandez, J. A. Sanz, H. Goyena, C.-T. Lin, Y. Wang, H. Bustince, Interval-valued aggregation functions based on moderate deviations applied to motor-imagery-based brain computer interface, IEEE Transactions on Fuzzy Systems (2021).

[15] M. Grabisch, J.-L. Marichal, R. Mesiar, E. Pap, Aggregation functions, Vol. 127, Cambridge University Press, 2009.

[16] H. Bustince, G. Beliakov, G. P. Dimuro, B. Bedregal, R. Mesiar, On the definition of penalty functions in data aggregation, Fuzzy Sets and Systems 323 (2017) 1–18.

[17] M. Vijay, A. Kashyap, A. Nagarkatti, S. Mohanty, R. Mohan, N. Krupa, Extreme gradient boosting classification of motor imagery using common spatial patterns, in: 2020 IEEE 17th India Council International Conference (INDICON), 2020, pp. 1–5.

[18] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun,



C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. Mueller-Putz, et al., Review of the bci competition iv, Frontiers in neuroscience 6 (2012) 55.

[19] A. Chowdhury, J. Andreu-Perez, Clinical brain-computer interface challenge 2020 (cbcic at wcci2020): Overview, methods and results, IEEE Transactions on Medical Robotics and Bionics (2021).

[20] M. Akin, Comparison of wavelet transform and fft methods in the analysis of eeg signals, Journal of medical systems 26 (3) (2002) 241–247.

[21] C. Vidaurre, T. Jorajuría, A. Ramos-Murguialday, K. R. Müller, M. Gómez, V. V. Nikulin, Improving motor imagery classification during induced motor perturbations, Journal of neural engineering 18 (4) (2021).

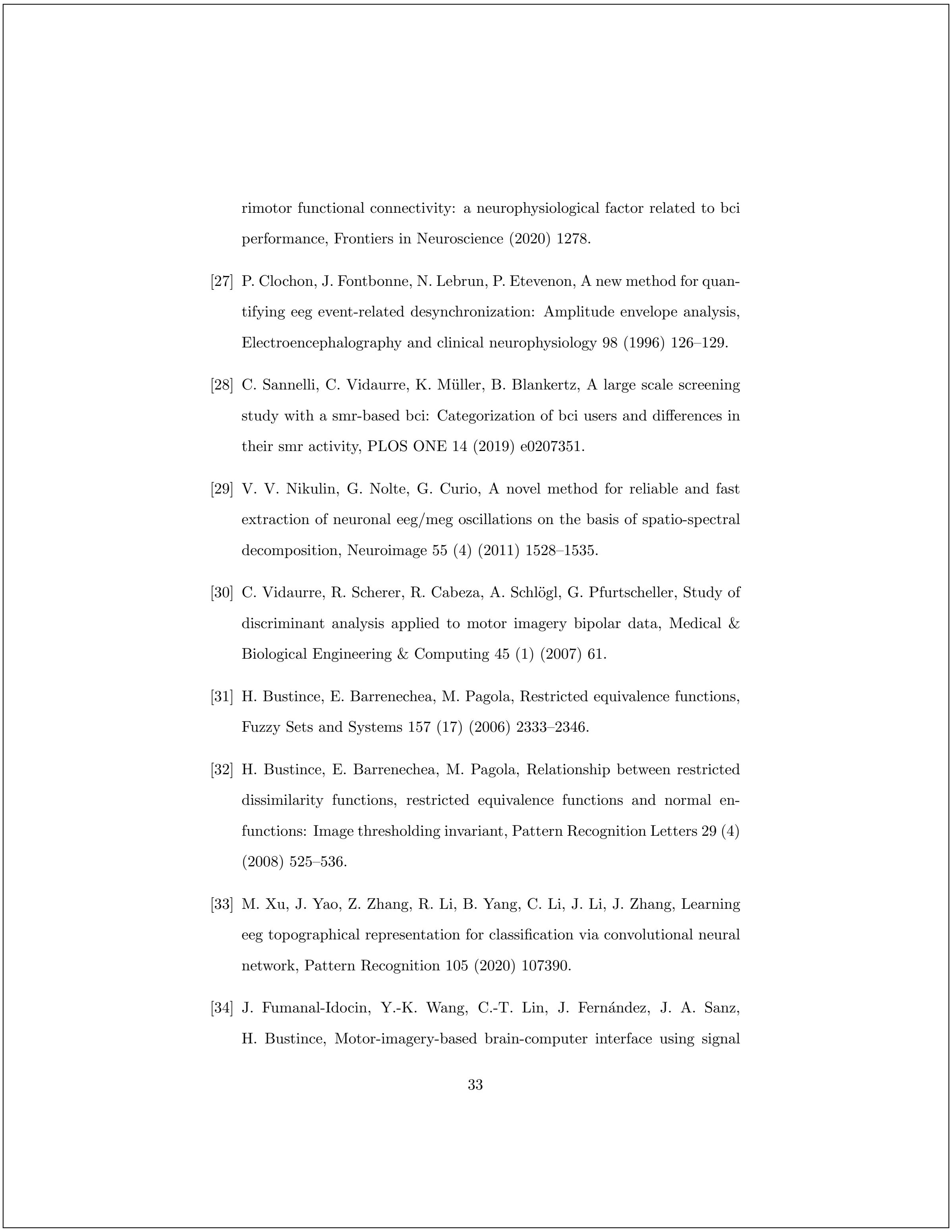
[22] J. Jin, R. Xiao, I. Daly, Y. Miao, X. Wang, A. Cichocki, Internal feature selection method of csp based on l1-norm and dempster–shafer theory, IEEE Transactions on Neural Networks and Learning Systems 32 (11) (2021) 4814–4825.

[23] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Classification of covariance matrices using a riemannian-based kernel for bci applications, Neurocomputing 112 (2013) 172–178.

[24] M. Hamedi, S.-H. Salleh, A. M. Noor, I. Mohammad-Rezazadeh, Neural network-based three-class motor imagery classification using time-domain features for bci applications, in: 2014 IEEE Region 10 Symposium, IEEE, 2014, pp. 204–207.

[25] A. J. Izenman, Linear discriminant analysis, in: Modern multivariate statistical techniques, Springer, 2013, pp. 237–280.

[26] C. Vidaurre, S. Haufe, T. Jorajuría, K.-R. Müller, V. V. Nikulin, Senso-



rimotor functional connectivity: a neurophysiological factor related to bci performance, Frontiers in Neuroscience (2020) 1278.

[27] P. Clochon, J. Fontbonne, N. Lebrun, P. Etevenon, A new method for quantifying eeg event-related desynchronization: Amplitude envelope analysis, Electroencephalography and clinical neurophysiology 98 (1996) 126–129.

[28] C. Sannelli, C. Vidaurre, K. Müller, B. Blankertz, A large scale screening study with a smr-based bci: Categorization of bci users and differences in their smr activity, PLOS ONE 14 (2019) e0207351.

[29] V. V. Nikulin, G. Nolte, G. Curio, A novel method for reliable and fast extraction of neuronal eeg/meg oscillations on the basis of spatio-spectral decomposition, Neuroimage 55 (4) (2011) 1528–1535.

[30] C. Vidaurre, R. Scherer, R. Cabeza, A. Schlögl, G. Pfurtscheller, Study of discriminant analysis applied to motor imagery bipolar data, Medical & Biological Engineering & Computing 45 (1) (2007) 61.

[31] H. Bustince, E. Barrenechea, M. Pagola, Restricted equivalence functions, Fuzzy Sets and Systems 157 (17) (2006) 2333–2346.

[32] H. Bustince, E. Barrenechea, M. Pagola, Relationship between restricted dissimilarity functions, restricted equivalence functions and normal en-functions: Image thresholding invariant, Pattern Recognition Letters 29 (4) (2008) 525–536.

[33] M. Xu, J. Yao, Z. Zhang, R. Li, B. Yang, C. Li, J. Li, J. Zhang, Learning eeg topographical representation for classification via convolutional neural network, Pattern Recognition 105 (2020) 107390.

[34] J. Fumanal-Idocin, Y.-K. Wang, C.-T. Lin, J. Fernández, J. A. Sanz, H. Bustince, Motor-imagery-based brain-computer interface using signal

derivation and aggregation functions, IEEE Transactions on Cybernetics (2021).

[35] A. Jafarifarmand, M. A. Badamchizadeh, S. Khanmohammadi, M. A. Nazari, B. M. Tazehkand, A new self-regulated neuro-fuzzy framework for classification of eeg signals in motor imagery bci, IEEE Transactions on Fuzzy Systems 26 (3) (2017) 1485–1497.

[36] A. S. Aghaei, M. S. Mahanta, K. N. Plataniotis, Separable common spatio-spectral patterns for motor imagery bci systems, IEEE Transactions on Biomedical Engineering 63 (1) (2015) 15–29.

[37] A. Natekin, A. Knoll, Gradient boosting machines, a tutorial, Frontiers in neurorobotics 7 (2013) 21.

[38] M. Hersche, T. Rellstab, P. D. Schiavone, L. Cavigelli, L. Benini, A. Rahimi, Fast and accurate multiclass inference for mi-bcis using large multiscale temporal and spectral features, in: 2018 26th European Signal Processing Conference (EUSIPCO), 2018, pp. 1690–1694.

[39] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, B. J. Lance, Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces, Journal of Neural Engineering 15 (5) (2018) 056013.

[40] S. R. Tibor, S. J. Tobias, F. L. D. Josef, G. Martin, E. Katharina, T. Michael, H. Frank, B. Wolfram, B. Tonio, Deep learning with convolutional neural networks for eeg decoding and visualization, Human Brain Mapping 38 (11) (2017) 5391–5420.

6.6 A Generalization of the Aggregation of Forces in the Gravitational Clustering to Perform Anomaly Detection

Associated publication:

- Fumanal-Idocin, J., Rodriguez-Martinez, I., Indurain, A., Minárová, M., & Bustince, H. (2022). A Generalization of the Aggregation of Forces in the Gravitational Clustering to Perform Anomaly Detection. *Information Sciences.*

  Status: Published.

  Impact Factor (JCR 2021): 8.233.

  Categories:

  Computer Science, Information Systems. Ranking 16/164 (Q1).

Almost Aggregations in the Gravitational Clustering to Perform Anomaly Detection

J. Fumanal-Idocin[a], I. Rodriguez-Martinez[a], A. Indurain[a], M. Minárová[b], C. Guerra[a], J. Fernandez[a], H. Bustince[a,*]

[a]*Universidad Pública de Navarra and Institute of Smart Cities, Campus Arrosadia s/n, 31006 Pamplona, Spain*
[b]*Slovak University of Technology in Bratislava, Radlinského 11, 81005 Bratislava, Slovakia*

Abstract

Anomaly detection is the process of identifying observations that differ significantly from the norm in a dataset. Since there is no proper formal definition of anomaly, different algorithms have arised to cope with the different variations of this idea, like novelty detection or outlier detection. Some of these algorithms have traditionally relied on prior knowledge on the data domain, or some degree of supervised learning in order to detect the irregular samples. In this work we propose a simulation based algorithm to detect those observations, based on gravitational forces, that requires neither prior knowledge nor data labels to identify spurious observations. We do so by studying different generalizations of the aggregation of gravitational forces, and the resulting clusters obtained when the particles attract each other. We also compare our algorithm with other unsupervised anomaly detection algorithms, obtaining favourable results to our proposal.

Keywords: Gravitational algorithm, Anomaly detection, Clustering, Aggregation functions

1. Introduction

Recognition of natural structures in data gives us more insight about the process behind the data itself [7]. Due to the rise of the user-generated content (i.e. tweets, Facebook posts, blogs, etc.) there are huge amounts of easily accessible, non-labeled data in a wide range of topics [22, 28]. It is also very common to find incomplete, missing or badly collected information [2]. Clustering is one of the most common tasks performed on non-labeled [37]. This process consists in assembling observations into different groups according to some criteria, so that objects inside one group are more

[*] Corresponding author
Email addresses: `javier.fumanal@unavarra.com` (J. Fumanal-Idocin), `iosu.rodriguez@unvarra.es` (I. Rodriguez-Martinez), `indurain.116020@e.unavarra.es` (A. Indurain), `maria.minarova@stuba.sk` (M. Minárová), `carlos.guerra@unavarra.com` (C. Guerra), `fcojavier.fernandez@unavarra.com` (J. Fernandez), `bustince@unavarra.es` (H. Bustince)

Preprint submitted to Elsevier *April 4, 2022*

similar to each other than those in other groups. There are a wide range of clustering algorithms: hierarchical clustering methods apply a similarity function, and form a tree-like structure by joining close points [39]; graph-based methods, use a graph to represent data similarity and looks for densely connected subgraphs [32, 6]; simulation-based algorithms are based on some kind of natural law, and adapt it and use it to look for optimal clusters [18]. Gravitational algorithms are part of this last group.

The main idea behind gravitational clustering is to simulate gravitational forces [36]. In this algorithm the observations are modelled as particles that attract each other due to an attractive gravitational force. They move closer to each other until a pair of particles gets close enough, then both of them collapse conforming a new one. This process repeats until there is only one particle left and then, the most stable configuration found is chosen as the final result. Starting from the original idea, there has been a wide range of algorithms following it. The authors in [29] present a fuzzy gravitational approach, where they select initial centres according to a density measure. In [12], the authors use gravitational theory to improve an existing algorithm and in [13] the authors use a generalization of the gravitational algorithm to detect communities in a social network.

One problem in data analysis is the presence of anomalies. An anomaly is an observation that is very different from the rest in a dataset and that may cause problems in the analysis of the sample. They might be present due to an error in the measure, faulty data or because the process that created the observation is different from the rest. There are different ways to detect anomalies. Density based approaches look for observations that are far from the rest of the points [9], [19]. Statistical models assume a distribution in the data and then use it to measure how likely is an observation to be an outlier [33]. Machine learning approaches treat this problem similarly to a classification problem [34], [25].

Gravitational clustering algorithms have been successfully applied in clustering problems, since the notion of gravitational force and the way it forms groups are easily understandable [31, 27, 8]. However, spurious data can result in problems, when the most stable configuration have been determined by a particle that was very far away from the rest. However, this also posses the chance to solve the anomaly detection problem using the opposite reasoning: we can detect anomalies in a dataset because they result in faulty results using gravitational or gravitational-like forces to form the groups.

We aim to exploit this possibility by presenting a generalization of the gravitational clustering algorithm, using aggregation functions instead of the product of masses and the summation of forces, in order to adapt the algorithm to anomaly detection. These functions have been widely used in a myriad of different problems regarding information fusion [38, 30, 17] and the Choquet and the Sugeno integrals are two of the most popular aggregation functions in literature [11, 26]. They have been successfully applied in Fuzzy Rules systems [24] and Brain Computer Interface Systems [20, 14, 16, 15], among others.

The contribution of this paper is twofold:

- To replace the sum by the Choquet integral, the Sugeno integral and some generalizations of the latter in order to get the resulting force in the gravitational clustering algorithm.

- To use this new version of the algorithm to adapt the gravitational clustering to an anomaly detection problem. In order to obtain the generalized version of the algorithm, we also replace the product by a more general class of functions which are increasing and symmetric.

We have tested our solution in a series of different real-world datasets adapted to the anomaly detection problem, comparing out proposal with the results of other anomaly detection approaches.

The rest of the paper goes as follows. In Section 2, we explain the basic concepts related to aggregation functions and the gravitational algorithm. In Section 3 we present the concept of almost aggregation, and in Section 4 we present our extension of the Choquet and Sugeno integrals for the vector space. In Section 5 we present our generalization for the gravitational algorithm. Subsequently, in Section 6 we discuss the changes to the system convergence when including the new aggregation functions. In Section 7 we report our results for different versions of the algorithm, and in Section 8 we compare them to other methods. Finally, in section 9 we summarize the whole work with some final remarks.

2. Preliminaries

We start recalling some notions that are necessary for the subsequent developments in this work.

2.1. *Aggregation functions.*

Let $n \geq 2$. An n-dimensional aggregation function on the unit interval is a function $M : [0,1]^n \to [0,1]$ such that [5]:

1. $M(0,\ldots,0) = 0$.
2. $M(1,\ldots,1) = 1$.
3. M is increasing in each variable.

Some examples of aggregation functions are, for instance:

- The minimum: $\min(x_1,\ldots,x_n)$ and the maximum $\max(x_1,\ldots,x_n)$.
- The product $P(x_1,\ldots,x_n) = x_1 \cdot \cdots \cdot x_n$.
- The arithmetic mean: $A(x_1,\ldots,x_n) = \frac{x_1+\cdots+x_n}{n}$.

2.2. *The Choquet and the Sugeno integrals*

The Choquet integral is an aggregation function that it is able to take the relationship which may exist between inputs into account [17]. The key notion behind the idea of Choquet integral is that of fuzzy measure, that we recall now. First of all, given $n \geq 1$, we denote $[n] = \{1,\ldots,n\}$ and the set of subsets of $[n]$ by $2^{[n]}$.

Definition 2.1. Let $n \geq 1$. A fuzzy measure over $[n]$ is a set-valued function $\mathfrak{m} : 2^{[n]} \to [0,1]$ such that:

- $\mathfrak{m}(\emptyset) = 0$ and $\mathfrak{m}([n]) = 1$;
- If $A \subseteq B \subseteq [n]$ then $\mathfrak{m}(A) \leq \mathfrak{m}(B)$.

Example 2.2. 1. In this work, the most important measure is that of cardinality (or uniform measure), which is defined, for any $A \subseteq [n]$, by:

$$\mathfrak{m}_1(A) = \frac{|A|}{n} \tag{1}$$

where $|A|$ denotes the cardinality (the number of elements) of the subset A.

2. Given any $q > 0$, another fuzzy measure is defined by:

$$\mathfrak{m}_q(A) = (\frac{|A|}{n})^q \tag{2}$$

which is also based on cardinality. Note that for $q = 1$ we recover the measure $\mathfrak{m}_1$.

3. Given $n_0 \in [n]$, the Dirac measure δ_{n_0} is defined for every $A \subseteq [n]$:

$$\delta_{n_0}(A) = \begin{cases} 1 & \text{if } n_0 \in A \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

A fuzzy measure $\mathfrak{m} : 2^{[n]} \to [0,1]$ is called symmetric if for any $A, B \subseteq [n]$ such that $|A| = |B|$ it holds that $\mathfrak{m}(A) = \mathfrak{m}(B)$. Note that $\mathfrak{m}_q$ is symmetric for any $q > 0$. However, the Dirac measure is not symmetric.

Now we can define the Choquet integral.

Definition 2.3. Let $n \geq 1$and let $\mathfrak{m} : 2^{[n]} \to [0,1]$ be a fuzzy measure. The Choquet integral (based on the measure $\mathfrak{m}$) is the function $Ch_{\mathfrak{m}} : [0,1]^n \to [0,1]$ defined, for every $(x_1, \ldots, x_n) \in [0,1]^n$, by:

$$Ch_{\mathfrak{m}}(x_1, \ldots, x_n) = \sum_{i=1}^{n} (x_{\sigma(i)} - x_{\sigma(i-1)}) \mathfrak{m}(A_{\sigma(i)}) \tag{4}$$

where

- $\sigma : [n] \to [n]$ is a permutation of the elements in $[n]$ such that $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(n)}$
- $A_{\sigma(i)} = \{x_{\sigma(i)}, x_{\sigma(i+1)}, \ldots, x_{\sigma(n)}\}$ for $i = 1, \ldots, n$ and $x_{\sigma(0)} = 0$.

For any fuzzy measure $\mathfrak{m} : 2^{[n]} \to [0,1]$, the Choquet integral $Ch_{\mathfrak{m}} : [0,1]^n \to [0,1]$ is an aggregation function [5].

Another example of aggregation function defined in terms of a fuzzy measure is the Sugeno integral [11].

Definition 2.4. Let $n \geq 1$and let $\mathfrak{m} : 2^{[n]} \rightarrow [0,1]$ be a fuzzy measure. The Sugeno integral (based on the measure $\mathfrak{m}$) is the function $S_{\mathfrak{m}} : [0,1]^n \rightarrow [0,1]$ defined, for every $(x_1, \dots, x_n) \in [0,1]^n$, by:

$$S_{\mathfrak{m}}(x_1, \dots, x_n) = \max_{i=1}^{n} \min(x_{\sigma(i)}, \mathfrak{m}(A_{\sigma(i)})) \tag{5}$$

with the same notations as in Def. 2.3.

For every fuzzy measure $\mathfrak{m}$, the function $S_{\mathfrak{m}} : [0,1] \rightarrow [0,1]$ is an aggregation function.

2.3. Algorithm of Gravitational Clustering

The gravitational clustering algorithm [35] employs the Newton gravitational law within the process of clustering. The scheme of this original algorithm is as follows. Assume that we have N particles $p_1, \dots, p_N$, with their positions $s_1, \dots, s_N \in \mathbb{R}^n$.

1. Initially we:
    - assign a mass 1/N to each particle p_i,
    - fix two real positive parameters ε and δ. We utilize δ for determining the actual time step longitude dt. Specifically, dt has to be such that during the time slot $[t, t+dt]$ the fastest particle displacement is equal to δ. Besides, we use ε to determine how close two particles can be before they are merged into one single particle, see step 2 (ii) below,
    - If in a moment two particles find themselves in a distance less than ε, we unify them into one single particle, with the mass equal to sum of masses of both of them and position done by their center of gravity.
    - Set initial time $t = 0$.
2. We repeat the following steps (i)-(iv) until one single particle remains.

    (i) In each time interval $[t, t+dt]$, for each particle i we compute its movement influencing function:

    $$g(i,t,dt) = \frac{1}{2} G \sum_{j \neq i} \frac{m_i(t) m_j(t)}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|^3} dt^2 \tag{6}$$

    where G is a positive constant.

    (ii) For each particle i, its new position is:

    $$s_i(t+dt) = s_i(t) + g(i,t,dt) \tag{7}$$

    (iii) We raise t to $t + dt$.

    (iv) If two particles i and j are in a distance less than ε, we unify them into one single particle, with its mass equal to sum of masses of both of them and its position given by their center of gravity of the two original particles.

Finally, we have just one particle. The relative life of the configuration with k clusters can be computed as

$$R_k = \frac{t_{k+1} - t_k}{T}$$

We choose as solution that corresponds to the configuration with largest relative life.

This model described above can be generalized by using a more general expression for particle movement governing function instead of (6). This was already done in [35] by using the following family of expressions

$$g(i,t,dt) = \frac{1}{2} G \sum_{j \neq i} \frac{m_i(t)^a m_j(t)^b}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|^3} dt^2 \tag{8}$$

with $a, b > 0$.

In particular, it was shown in [36] that, from an experimental point of view, the best results are obtained for $a = b = 0$. the resulting gravitational model is called Markov unitary model and it is described by the governing function:

$$g(i,t,dt) = \frac{1}{2} G \sum_{j \neq i} \frac{1}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|^3} dt^2 \tag{9}$$

3. Almost aggregation functions

In this work, it will be necessary to fuse data that are not necessarily in the interval $[0,1]$. For this sake, we will establish a notion of almost aggregation function operating over interval $[0,\infty[$.

Definition 3.1. An n-dimensional function $M : ([0,\infty[)^n \to [0,\infty[$ for $n \geq 2$ will be called almost aggregation function on the interval $[0,\infty[$ if:

1. $M(0,\ldots,0) = 0$.
2. M is increasing in each variable (usual monotonicity).

Note that in Def. 3.1 we do not impose a boundary condition at ∞. For this reason, we do not speak of aggregation function. However, all the previous examples of aggregation functions defined in $[0,1]^n$ can be straightforwardly extended to be almost aggregation functions on $[0,\infty[^n$.

In particular, observe that, if we take any fuzzy measure $\mathfrak{m}$ on $[n]$, the function $Ch_{\mathfrak{m}} : ([0,\infty[)^n \to [0,\infty[$, defined as:

$$Ch_{\mathfrak{m}}(x_1,\ldots,x_n) = \sum_{i=1}^{n} (x_{\sigma(i)} - x_{\sigma(i-1)}) \mathfrak{m}(A_{\sigma(i)}) \tag{10}$$

where

- $\sigma : [n] \to [n]$ is a permutation of the elements in $[n]$ such that $x_{\sigma(1)} \leq x_{\sigma(2)} \leq \cdots \leq x_{\sigma(n)}$.
- $A_{\sigma(i)} = \{x_{\sigma(i)}, x_{\sigma(i+1)}, \ldots, x_{\sigma(n)}\}$ for $i = 1,\ldots,n$ and $x_{\sigma(0)} = 0$.

is an almost aggregation function on $([0,\infty[)^n$, and this is also the case for the Sugeno integral. With respect to the generalized Sugeno integral in Def. 2.4, we also recover an almost aggregation function if we consider inputs in $[0,\infty[$.

In order to get other examples of almost aggregation functions on the interval $[0,\infty[$, we can generalize the Sugeno integral as follows [4].

Definition 3.2. Let $n \geq 2$. Let $\mathfrak{m} : 2^{[n]} \to [0,1]$ be a symmetric fuzzy measure, $F : [0,\infty[\times[0,1] \to [0,\infty[$ a bivariate function and $G : ([0,\infty[)^n \to [0,\infty[$ an n-ary function. A Sugeno-like FG-function is a function $A : ([0,\infty[)^n \to [0,\infty[$ given by:

$$A(x_1,\ldots,x_n) = G(F(x_{\sigma(1)},\mathfrak{m}(A_{(1)})),\ldots,F(x_{\sigma(n)},\mathfrak{m}(A_{(n)})) \tag{11}$$

for every $(x_1,\ldots,x_n) \in ([0,\infty[)^n$, with σ and $A_{(\cdot)}$ the same as in Def. 2.3.

If $F = \min$ and $G = \max$, we recover the usual Sugeno integral. For this work, we are specially interested in considering $F = \min$ and $G = \sum$, which gives back an almost aggregation function on $[0,\infty[$.

4. Modification of the aggregation of attraction forces

In this section we present the changes performed to the gravitational attraction formula: substituting the product of the masses for a new kind of functions, that we call H functions, and replacing the summation of forces by the Choquet integral, the Sugeno Integral a generalization of the Sugeno integral . Along the remainder of the paper, for a fixed time $t > 0$, by abuse of notation, we denote the number of particles $N(t)$ at the instant t just by N.

The modification that we intend to do require some additional considerations. Note that both Choquet and Sugeno integrals, as well as the generalizations of the latter, are defined for aggregating numbers. However, the attraction forces are vectors, so it is necessary to adapt both integrals to deal with vectorial data. We propose in the following subsections an ad-hoc version of these integrals to apply in this algorithm, that uses the attraction force vector module to do the sorting of these vectors.

First of all, we need to introduce a new class of functions that we will use to generalize the product in the expression of the force.

4.1. H functions

Observe that the product of the masses in the gravitational algorithm always involves two integer masses. In order to generalize this discrete product, given an integer $N \geq 1$, we are going to take a function: $H\{1,\ldots,N\} \times \{1,\ldots,N\} \to \mathscr{H} \subset \mathscr{R}$, such that:

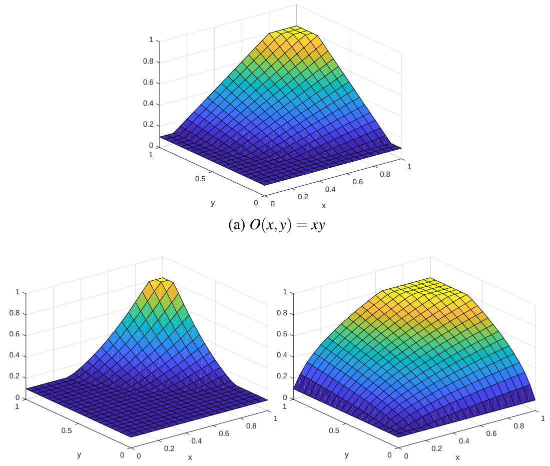
- H is increasing.
- H is symmetric.
- H is bounded on $\{1,\ldots,N\} \times \{1,\ldots,N\}$.

This function holds similar properties to overlap functions [10], with no conditions when $xy = 0$ or $xy = 1$. So, taking into account that $(xy)^c$ is an overlap function for any $c > 0$, we consider the following specific case of H functions:

$$H_c(x, y) = \begin{cases} B_1 & \text{if } (xy)^c < B_1, \\ B_2 & \text{if } (xy)^c > B_2, \\ (xy)^c & \text{if } B_1 \leq (xy)^c \leq B_2 \end{cases} \tag{12}$$

where $0 \leq B_1 \leq B_2$ and $c > 0$ is a real parameter of the function.

In Figure 1 we plot the function H_c for different values of c. Of course , other overlap functions $O : [0, 1]^2 \to [0, 1]$ can be used in the definition H instead of $(xy)^c$.



(a) $O(x, y) = xy$

(b) $O(x, y) = (xy)^2$

(c) $O(x, y) = \sqrt{xy}$

Figure 1: Visualizations of the H functions(Eq. (12)) for different overlaps with parameters $B_1 = 0.1$ and $B_2 = 0.8$.

4.2. *Computing the Choquet Integral Aggregation of Forces*

Let us fix a time $t > 0$. In the time instant t the particle p_i is driven by the acceleration given in Eq. 6. This expression can be seen as a weighted sum of vectors. From this point of view, and taking into account that the weighted sum of numbers can be recovered as a specific instance of the Choquet integral, we intend to consider the latter to get a more general expression. To do so, we need to order the inputs, and, as we have already said, we are going to do this ordering by using the force moduli. In this way, the new aggregation of forces can be computed as follows:

1. For each p_i in time t compute the sequence of force moduli and assemble the values in a vector $\vec{v}(t)$, with components $v_j(t)$:
$$v_j(t) = \frac{H_c(m_j(t), m_i(t))}{|\vec{s}_j(t) - \vec{s}_i(t)|^2} \tag{13}$$
2. Sort the moduli in non-decreasing way. That is, consider a permutation σ of $[n]$ such that:
$$\frac{H_c(m_{\sigma(1)}(t), m_i(t))}{|\vec{s}_{\sigma(1)}(t) - \vec{s}_i(t)|^2} \leq \frac{H_c(m_{\sigma(2)}(t), m_i(t))}{|\vec{s}_{\sigma(2)}(t) - \vec{s}_i(t)|^2} \leq ... \leq \frac{H_c(m_{\sigma(N-1)}(t), m_i(t))}{|\vec{s}_{\sigma(N-1)}(t) - \vec{s}_i(t)|^2} \tag{14}$$
3. Taking $v_0(t) = 0$, acquire the components $v_j(t) - v_{j-1}(t)$ of an $(N-1)$-tuple Choquet-like integral vector. Multiply each component of Choquet-like integral vector by corresponding measure $h_{\sigma(j)}$, yielding the Choquet-like integral weighting vector
$$\vec{Ch}_i(t)^T = \left[\frac{H_c(m_{\sigma(1)}(t), m_i(t))}{|\vec{s}_{\sigma(1)}(t) - \vec{s}_i(t)|^2} h(A_{\sigma(1)}),\right.$$
$$\left(\frac{H_c(m_{\sigma(2)}(t), m_i(t))}{|\vec{s}_{\sigma(2)}(t) - \vec{s}_i(t)|^2} - \frac{H_c(m_{\sigma(1)}(t), m_i(t))}{|\vec{s}_{\sigma(1)}(t) - \vec{s}_i(t)|^2}\right) h(A_{\sigma(2)}), ...$$
$$\left. ..., \left(\frac{H_c(m_{\sigma(N-1)}(t), m_i(t))}{|\vec{s}_{\sigma(N-1)}(t) - \vec{s}_i(t)|^2} - \frac{H_c(m_{\sigma(N-2)}(t), m_i(t))}{|\vec{s}_{\sigma(N-2)}(t) - \vec{s}_i(t)|^2}\right) h(A_{\sigma(N-1)})\right] =$$
If the cardinality is used as the fuzzy measure h in Choquet-like integral weighting vector, we get
$$\vec{Ch}_i(t)^T = \left[\frac{H_c(m_{\sigma(1)}(t), m_i(t))}{|\vec{s}_{\sigma(1)}(t) - \vec{s}_i(t)|^2}, \left(\frac{H_c(m_{\sigma(2)}(t), m_i(t))}{|\vec{s}_{\sigma(2)}(t) - \vec{s}_i(t)|^2} - \frac{H_c(m_{\sigma(1)}(t), m_i(t))}{|\vec{s}_{\sigma(1)}(t) - \vec{s}_i(t)|^2}\right) \frac{N-2}{N-1}, ...\right.$$
$$\left. ..., \left(\frac{H_c(m_{\sigma(N-1)}(t), m_i(t))}{|\vec{s}_{\sigma(N-1)}(t) - \vec{s}_i(t)|^2} - \frac{H_c(m_{\sigma(N-2)}(t), m_i(t))}{|\vec{s}_{\sigma(N-2)}(t) - \vec{s}_i(t)|^2}\right) \frac{1}{N-1}\right]$$
Remark: $\vec{Ch}_i(t)$ with its $N-1$ components $\vec{Ch}_{i,\sigma(j)}(t)$ is specific for each particle p_i.
4. By using $(\sigma(1), ..., \sigma(N-1))$, permute, the line vectors of distances, one below the other, getting the distance matrix $\mathbb{D}_i(t)$ with dimension $(N-1) \times D$.

$$\mathbb{D}_i = \begin{bmatrix} \frac{\vec{s}_{\sigma(1)} - \vec{s}_i}{|\vec{s}_{\sigma(1)} - \vec{s}_i|} \\ \frac{\vec{s}_{\sigma(2)} - \vec{s}_i}{|\vec{s}_{\sigma(2)} - \vec{s}_i|} \\ ... \\ \frac{\vec{s}_{\sigma(N-1)} - \vec{s}_i}{|\vec{s}_{\sigma(N-1)} - \vec{s}_i|} \end{bmatrix} = \begin{bmatrix} \frac{s^1_{\sigma(1)} - s^1_i}{|\vec{s}_{\sigma(1)} - \vec{s}_i|} & \frac{s^2_{\sigma(1)} - s^2_i}{|\vec{s}_{\sigma(1)} - \vec{s}_i|} & ... & \frac{s^D_{\sigma(1)} - s^D_i}{|\vec{s}_{\sigma(1)} - \vec{s}_i|} \\ \frac{s^1_{\sigma(2)} - s^1_i}{|\vec{s}_{\sigma(2)} - \vec{s}_i|} & \frac{s^2_{\sigma(2)} - s^2_i}{|\vec{s}_{\sigma(2)} - \vec{s}_i|} & ... & \frac{s^D_{\sigma(2)} - s^D_i}{|\vec{s}_{\sigma(2)} - \vec{s}_i|} \\ ... & & & \\ \frac{s^1_{\sigma(N-1)} - s^1_i}{|\vec{s}_{\sigma(N-1)} - \vec{s}_i|} & \frac{s^2_{\sigma(N-1)} - s^2_i}{|\vec{s}_{\sigma(N-1)} - \vec{s}_i|} & ... & \frac{s^D_{\sigma(N-1)} - s^D_i}{|\vec{s}_{\sigma(N-1)} - \vec{s}_i|} \end{bmatrix} \tag{15}$$

5. Perform the weighted aggregation of distances, i.e. multiply transposed $\vec{Ch}_i$ by $\mathbb{D}_i(t)$ and include it in the acceleration vector function; i.e. the vector function driving the movement of particle within the current time step.

$$\vec{g}_i(t) = \frac{1}{m_i(t)} \vec{Ch}_i(t)^T \mathbb{D}_i(t) dt^2 \tag{16}$$

Apparently, we get (16) from (21), for $\vec{w}_i(t)^T = \vec{Ch}_i(t)^T$ and $\mathbb{M}_i(t) = \mathbb{D}_i(t)$;

4.3. Computing the Sugeno Integral Aggregation of Forces

An analogous procedure can be developed if we want to use the Sugeno instead of the Choquet.

1. $\forall p_i$ in time t compute the moduli of all forces acting between p_i and each other particle; and record the moduli in $\vec{v}(t)$, with components $v_j(t)$.
2. Arrange the moduli nondecreasingly, (14). Remember the permutation $\sigma(j)$ in accordance with the definition 2.3
3. For each $\sigma(j)$ put

$$u_{\sigma(j)}(t) = \min\left(v_{\sigma(j)}, h(A_{\sigma(j)})\right) \tag{17}$$

and record in vector $\vec{u}_i(t)$

4. Find maximal component of $\vec{u}(t)$ and remember its argument

$$S_i(t) = \max_{\sigma(j)} u_{\sigma(j)}(t) \tag{18}$$

$$\alpha_i(t) = \arg(\max_{\sigma(j)} u_{\sigma(j)}(t)) \tag{19}$$

5. $\alpha_i{}^{th}$ particle is the only one influencing the current movement of i^{th} particle

$$\vec{g}_i(t) = \frac{1}{m_i(t)} S_i(t) \frac{\vec{s}_{\alpha(i)}(t) - \vec{s}_i(t)}{|\vec{s}_{\alpha(i)}(t) - \vec{s}_i(t)|} dt^2 \tag{20}$$

Let us recall that (20) can be regarded as a special case of (21) - when as $\vec{w}_i(t)^T$ an D-tuple vector $(S_i(t), ..., S_i(t))$ is taken, and the D dimensional square matrix $\mathbb{M}_i(t) = diag(\vec{d}_{\alpha(i)}(t))$ is put. Herein the diagonal elements are components of $\vec{d}_{\alpha(i)}(t) = \frac{\vec{s}_{\alpha(i)}(t) - \vec{s}_i(t)}{|\vec{s}_{\alpha(i)}(t) - \vec{s}_i(t)|}$. The generalized versions of the Sugeno integral the Sugeno-Hamacher and the FG-Sugeno, can be computed following the same procedure, changing the min-max function for the appropriate functions.

4.4. Computing the generalized Sugeno Integral Aggregation of Forces

1. Compute all moduli of all forces acting between p_i and each other particle; and record the moduli in $\vec{v}(t)$, with components $v_j(t)$.
2. Arrange the moduli non-decreasingly. Remember the used permutation
3. For each force modulus and corresponding fuzzy measure (cardinality) carry out the F function. Arrange all items into generalized Sugeno-like weighting vector
4. Involve product of the weighting vector and distance matrix in the acceleration vector function. Herein, the weighting vector is either FG-Sugeno or H-Sugeno like integral.

5. New gravitational clustering algorithm

Summarizing the theoretical developments in nthe previous section, our idea for the new clustering algorithm is to replace the expression $m_i(t)^{a-1}m_j(t)^b$ in Eq. (8) by a more general expression of the type $H(m_i^{a-1}, m_j^b)$ that can include both the product and the unitary model and any other possibly interesting functions. We also intend to adapt this formula to the different conditions throughout the execution of the algorithm.

We also change the sum in the original expression by an aggregation function. The benefit of an aggregation function is that we can weight each attraction force according to its module. Accordingly, we can treat small forces, which can be related to huge distances, i.e. anomalies. We have decided to use the Sugeno and the Choquet integrals, and their generalizations as almost aggregation functions.

We also want to make our algorithm being able to detect anomaly clusters. We call anomaly clusters those clusters in the most stable configuration that are very small and that lived a significant amount of time. To modulate those two conditions, we added two more new parameters to the algorithm: P, that establishes the size boundary between anomaly and non-anomaly clusters, and PT, which sets the minimum time that a cluster must hold in time to declare anomaly.

5.1. Dynamic aggregation of masses

Sometimes it is not possible to obtain a good result fixing B_1 and B_2 from the beginning. Usually, this happens when one or more outliers create a "fake" most stable configuration because they are "too far away" from the rest of data. The dynamic model tries to fix this problem by recalculating B_1 and B_2 in each iteration, aiming to reduce or augment the velocity. In doing so, we try to make outliers move as fast as possible and to make potentially good configurations more stable. If the dynamic version is used, then we can set initially B_1 and B_2 to 0 and 1 respectively, as they will be updated accordingly during the execution of the algorithm.

To achieve that, we slow down the particles at the beginning, until some condition is met. Then we speed up the process. This makes initial clusters more stable compared to latter combinations, so outliers that are too far from inliers will not take as much simulated time as in the static model. When to slow and when to speed things up - this is done heuristically. We check the biggest particle, the percentage of the total mass that it possesses, and compare it to a constant K between 0 and 1. If it is smaller, we slow down the process by reducing B_2, if bigger, we speed up the particles by increasing B_1.

For our experiments, we found the most appropriate K value tends to be around $0.10-0.15$.

5.2. Formulation of the new Gravitational Algorithm involving the distance aggregations for anomaly detection

When a configuration of N particles is given: $p_1, \ldots, p_N$, and their positions $s_1, \ldots, s_N \in \mathbb{R}^n$, the new algorithm that we propose is the following:

1. Initially, we:
    - Initially one particle p_i is one cluster with its mass equal to $1/N$.

- We fix real positive parameters ε and δ, $\delta < \varepsilon$ if the clustering is prior to attracting.
- We utilize δ for determining the actual time step longitude of the actual $dt = dt(t,\delta)$. Since δ is fixed, we have just $dt(t)$. Indeed, $[t, t + dt(t)]$ is the time period in which the most rapid particle moves by δ.
- If two particles find themselves in a distance less than ε, we unify them in one with the mass equal to the sum of masses of both of them and position in their center of gravity.
- We set $t = 0$ initially.
- Choose both an aggregation function, D, and a c for the H_c function.
- Set the K value to regulate the velocity in the clustering process.

2. We repeat following steps (i)-(iv) until the unique particle remains.

   (i) In each time interval $[t, t + dt]$, for each particle i we compute its movement influencing function:

   $$\vec{g}_i(t) = \frac{1}{m_i(t)} \vec{w}_i(t)^T \mathbb{M}_i(t) dt^2 \qquad (21)$$

   with $\vec{w}_i(t)$ being a vector of weights and $\mathbb{M}_i(t)$ distance matrix.

   (ii) We find the fastest particle

   $$I = \arg(\max_i \{|\vec{g}_i|(t)\}), \qquad (22)$$

   then we gain the length $dt(t)$ of current time step

   $$|\vec{g}_I(t)| = \delta \Rightarrow dt(t) \qquad (23)$$

   and finally the new position of each particle i is:

   $$\vec{s}_i(t + dt(t)) = \vec{s}_i(t) + \vec{g}_i(t) \qquad (24)$$

   (iii) We raise t to $t + dt(t)$.

   (iv) If two particles i and j are in a distance less than ε, we unify them.

   (iv) If the biggest mass is bigger than K, we raise B_1. If not, we decrease B_2.

   (iv) If two particles i and j are in a distance less than ε, we unify them.

Finally we have just one particle. For the sake of further use, we record the time points in a vector $\vec{\tau}$, when due to merging of particles, the number of particles decreases $\vec{\tau} = \{t_{N-k_1}, t_{N-k_2}, \ldots, t_{N-k_l} = t_1\}$, each index $0 < k_1 < \ldots < k_q < N$ standing as a numbers of remaining particles, since the number of particles can decrease by more than one in one step; e.g., within the interval $[t_N = 0, t_{k_1}]$ the number of particles is N, within the time interval $[t_{N-k_1}, t_{N-k_2}]$ the number of particles is $N - k_1$, etc. Formally, in $[t_1, \infty]$ all data are gathered in one cluster, i.e. one particle remains and persists.

The period of one level of a clustering configuration can be quantified relatively as

$$R_k = \frac{t_{N-k_l} - t_{N-k_{l+1}}}{T} \tag{25}$$

where T is the lifespan of entire clustering mechanism. Then, we mark as anomalies those particles that exited in clusters of size P or less that stayed alive less than the PT parameter.

When applying the model to a new observation, we use a 1-NN classifier, so that the new particle will be classified depending on its closer particle using the Euclidean distance.

5.3. Decision Making Scheme for the Generalized Gravitational Clustering for Anomaly Detection

We can use a similar strategy to bootstrap aggregation for our algorithm in order improve the results. This process consists of sampling different percentages of attributes and samples from the original dataset, and then using an aggregation function to compute the threshold to declare each particle an anomaly or not:

1. We set the size of the ensemble as E, and choose an aggregation function F_1.
2. For each model $e \in E$:
   - (2.1) We randomly chose a subset of features from the dataset.
   - (2.2) We randomly subsample the dataset.
   - (2.3) We compute the algorithm as in Section 5.2.
3. We denote the result for each model in the ensemble E for particle p as O_p. For each particle, we compute its outlier score, S as:

$$S = \frac{\sum O_p}{|E|} \tag{26}$$

4. The final threshold, T, is computed as:

$$T = F_1(\mathbf{S}) \tag{27}$$

   where $\mathbf{S}$ is the vector with the S for each individual particle.
5. We denote as anomalies all of those that have an outlier score bigger than T.

6. Convergence and shrinkage of the system in time

Ensuring the convergence of the algorithm involving the aggregation of distances means ensuring the contractivity of the system of particles, driven by the acceleration function (21), in time. In accordance with [3], we immerse the system of particles to a rectangular (convex) hull and control the particles in order to prevent them of escaping from the region, or remaining on its border. As the maximal shift within one time step is restricted to δ, we just have to check and control the particles in the distance δ or lower, from each border. If we take $D = 2$, we can illustrate the problem graphically (see Figure 2).

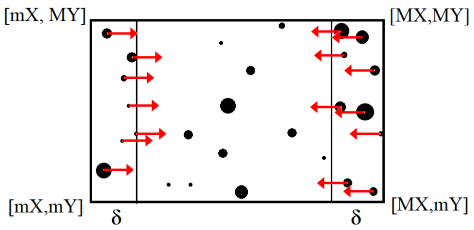


Figure 2: Particles' x coordinate control on the covering rectangular region. All particles in the δ margin are driven inwards.

6.1. Acceleration function with Choquet-like integral vector inside

Without loss of generality, we continue with two dimensions in order to have an illustration tool at hand. When we multiply the line vector $\vec{Ch}_i$ by matrix $\mathbb{D}_i$, as performed in (16), indeed we multiply the $\sigma(j)^{th}$ component of $\vec{Ch}_i(t)$ by the $\sigma(j)^{th}$ normalized position difference vector $\frac{\vec{s}_{\sigma(j)}(t)-\vec{s}_i(t)}{|\vec{s}_{\sigma(j)}(t)-\vec{s}_i(t)|}$ and carry out the summation through $\sigma(j)$. Accordingly, having the first column of matrix $\mathbb{D}_i(t)$ assembled in vector $\vec{d}_i^1(t)$, the first coordinate from (16) can be withdrawn:

$$g_i^1(t) = \frac{1}{m_i(t)} \vec{Ch}_i(t)^T . \vec{d}_i^1(t) dt^2 \tag{28}$$

Next we follow the consideration exploiting the physical interpretation provided in [3]. As far as the first coordinate (x) concerns, we have to prevent the particles from both right and left hand δ margin of getting on the vertical borderline or even escaping out from the covering rectangle through it (Figure 2). We direct these particles inwards, hence the first coordinate of its acceleration has to be negative for the particles at the right hand side margin; and positive at the left hand side. Let us take a particle from the right hand side δ margin and assign index R to it. It has to read $g_R^1 < 0$, so

$$\frac{1}{m_R}(t)(Ch_{R,1}(t)d_{R,1}^1 + \ldots + Ch_{R,N-1}(t)d_{R,N-1}^1)dt^2(t) < 0 \tag{29}$$

and this is valid if and only if the sum in brackets is negative. Directly from the construction of the Choquet-like integral vector follows that all components of $\vec{Ch}_R(t)$ are non-negative. Let us rewrite this inequality in sense of H function:

$$\sum_{j=1}^{N-1} \left(\frac{H_c(m_{\sigma(j)}(t), m_R(t))}{|\vec{s}_{\sigma(j)}(t) - \vec{s}_R(t)|^2} - \frac{H_c(m_{\sigma(j-1)}(t), m_R(t))}{|\vec{s}_{\sigma(j-1)}(t) - \vec{s}_R(t)|^2} \right) \frac{N-j}{N-1} \frac{s_{\sigma(j)}^1(t) - s_R^1(t)}{|\vec{s}_{\sigma(j)}(t) - \vec{s}_R(t)|} < 0 \tag{30}$$

After truncating $\frac{1}{N-1}$ from (30) and abbreviating of the non-negative member

$$P_{\sigma(j)} = \left(\frac{H_c(m_{\sigma(j)}(t), m_R(t))}{|\vec{s}_{\sigma(j)}(t) - \vec{s}_R(t)|^2} - \frac{H_c(m_{\sigma(j-1)}(t), m_R(t))}{|\vec{s}_{\sigma(j-1)}(t) - \vec{s}_R(t)|^2} \right) (N-j)$$

we have the required convergence criterion in the form

$$\sum_{j=1}^{N-1} P_{\sigma(j)} \frac{s^1_{\sigma(j)}(t) - s^1_R(t)}{|\vec{s}_{\sigma(j)}(t) - \vec{s}_R(t)|} < 0 \tag{31}$$

Herein, $P_{\sigma(j)}$ can be regarded physically, too. The particles closer to particle p_R are more influencing for its next movement. Finally, (31) can be expressed in a form that reflects the split of all particles to those more and less far from the border than p_R. The particles with the same x coordinate have no influence on the movement of p_R in the x direction. For the sake of more clarity we omit independent variable (t) in setting up algorithm convergence criterion:

$$\sum_{j=1}^{N-1} P_{\sigma(j)} \frac{s^1_{\sigma(j)} - s^1_R + |s^1_{\sigma(j)} - s^1_R|}{2|\vec{s}_{\sigma(j)} - \vec{s}_R|} < \sum_{j=1}^{N-1} P_{\sigma(j)} \frac{s^1_R - s^1_{\sigma(j)} + |s^1_{\sigma(j)} - s^1_R|}{2|\vec{s}_{\sigma(j)} - \vec{s}_R(t)|} \tag{32}$$

This form of inscription emphasizes the fact that total attracting impact of the particles nearer to x-coordinate borderline has to be smaller than the impact of all others.
Remark: The convergence criterion subsequently determines setting delta parameter of the algorithm.

6.2. *Acceleration function with Sugeno-like integral vector inside*

Likewise in case of Choquet-like integral vector, we work in two dimensions; with regard of future easy enhancing to D dimensions. Form (20) we can withdraw the first coordinate of the acceleration vector and control the x coordinate of the movement of the particle p_R situated on the right hand side δ margin of the domain,

$$g^1_R(t) = \frac{1}{m_R(t)} S_R(t) \frac{\vec{s}_{\alpha(R)}(t) - \vec{s}_R(t)}{|\vec{s}_{\alpha(R)}(t) - \vec{s}_R(t)|} \tag{33}$$

demanding its negativity. Since $S_R > 0$, equivalently, we require

$$s^1_{\alpha(R)}(t) - s^1_R(t) < 0 \iff s^1_{\alpha(R)}(t) < s^1_R(t) \tag{34}$$

This means δ parameter should be set so that α^{th} particle will not fall into a δ margin of the covering domain. And it is not possible to ensure.

6.3. *Acceleration function with FG-Sugeno-like integral vector inside*

Let us take $G = \sum$ and $F(x,y) = x.y$, in FG-Sugeno integral, see definition 2.4. We have

$$F(H_c(m_{\sigma(j)}, m_i), \frac{N-j}{N-1}) = \frac{H_c(m_{\sigma(j)}, m_i)(N-j)}{N-1} \tag{35}$$

Herein, the acceleration vector governing the particle p_i movement is

$$\vec{g}_i(t) = \frac{1}{m_i(t)} \vec{V}_i(t)^T \mathbb{D}_i dt^2 \tag{36}$$

and its first coordinate is then

$$g_i^1 = \frac{1}{m_i}\sum_{j=1}^{N-1} H_c(m_{\sigma(j)},m_i)\frac{N-j}{N-1}\frac{s^1_{\sigma(j)}-s^1_i}{|\vec{s}_{\sigma(j)}-\vec{s}_R|}dt^2 \tag{37}$$

As reasoned above, at the right hand side δ margin of covering rectangle we require negativeness of the g^1_R, the 1^{st} coordinate of the acceleration of particles. $g^1_R < 0 \iff$:

$$\sum_{j=1}^{N-1} H_c(m_{\sigma(j)},m_R)(N-j)\frac{s^1_{\sigma(j)}-s^1_R}{|\vec{s}_{\sigma(j)}-\vec{s}_R|} < 0 \tag{38}$$

and its split reflecting the position of the particles yields:

$$\begin{aligned}\sum_{j=1}^{N-1} H_c(m_{\sigma(j)},m_R)(N-j)\frac{s^1_{\sigma(j)}-s^1_R+|s^1_{\sigma(j)}-s^1_R|}{|\vec{s}_{\sigma(j)}-\vec{s}_R|} < \\ < \sum_{j=1}^{N-1} H_c(m_{\sigma(j)},m_R)(N-j)\frac{s^1_R-s^1_{\sigma(j)}+|s^1_{\sigma(j)}-s^1_R|}{|\vec{s}_{\sigma(j)}-\vec{s}_R|}\end{aligned} \tag{39}$$

6.4. Acceleration function with H-Sugeno-like integral vector inside

G=sum and $F(x,y) = \frac{xy}{x+y-xy}$

$$F(H_c(m_{\sigma(j)},m_i),\frac{N-j}{N-1}) = \frac{H_c(m_{\sigma(j)},m_i)\frac{N-j}{N-1}}{H_c(m_{\sigma(j)},m_i)+\frac{N-j}{N-1}-H_c(m_{\sigma(j)},m_i)\frac{N-j}{N-1}}$$

We have acceleration of the form

$$\vec{g}_i(t) = \frac{1}{m_i(t)}\vec{W}_i(t)^T\mathbb{D}_i dt^2 \tag{40}$$

and its first coordinate

$$g_i^1 = \frac{1}{m_i}\sum_{j=1}^{N-1}\frac{H_c(m_{\sigma(j)},m_i)\frac{N-j}{N-1}}{H_c(m_{\sigma(j)},m_i)+\frac{N-j}{N-1}-H_c(m_{\sigma(j)},m_i)\frac{N-j}{N-1}}\frac{s^1_{\sigma(j)}-s^1_i}{|\vec{s}_{\sigma(j)}-\vec{s}_R|}dt^2 \tag{41}$$

Likewise in above cases, we require negativeness of the g^1_R, the 1^{st} coordinate of the acceleration of particles at the right hand side δ margin of covering rectangle. $g^1_R < 0 \iff$:

$$\sum_{j=1}^{N-1}\frac{H_c(m_{\sigma(j)},m_R)(N-j)}{(N-1)H_c(m_{\sigma(j)},m_R)+(N-j)-H_c(m_{\sigma(j)},m_R)(N-j)}\frac{s^1_{\sigma(j)}-s^1_R}{|\vec{s}_{\sigma(j)}-\vec{s}_R|} < 0$$

and by rearranging of this form we have sufficient criterion for convergence on the right hand side of the covering rectangle

$$\sum_{j=1}^{N-1}\frac{H_c(m_{\sigma(j)},m_R)(N-j)}{(j-1)H_c(m_{\sigma(j)},m_R)+N-j}\frac{s^1_{\sigma(j)}-s^1_R}{|\vec{s}_{\sigma(j)}-\vec{s}_R|} < 0 \tag{42}$$

and its split reflecting the position of the particles yields:

$$\sum_{j=1}^{N-1} \frac{H_c(m_{\sigma(j)}, m_R)(N-j)}{(j-1)H_c(m_{\sigma(j)}, m_R) + N - j} \frac{s^1_{\sigma(j)} - s^1_R + |s^1_{\sigma(j)} - s^1_R|}{|\vec{s}_{\sigma(j)} - \vec{s}_R|} <$$

$$< \sum_{j=1}^{N-1} \frac{H_c(m_{\sigma(j)}, m_R)(N-j)}{(j-1)H_c(m_{\sigma(j)}, m_R) + N - j} \frac{s^1_R - s^1_{\sigma(j)} + |s^1_{\sigma(j)} - s^1_R|}{|\vec{s}_{\sigma(j)} - \vec{s}_R|} \tag{43}$$

(43) says that the weight of the influence of the particles attracting the particle outward in x direction has to be lower than the influence of the other particles.

7. Experimental Results

In this section we present the results obtained using the different parameters of our proposed gravitational algorithm. For our experimentation, we have used five different datasets, obtained from the KEEL repository [1], which are summarized in Table 1. These datasets were originally proposed for classification and the preprocessing required to convert each dataset into an anomaly detection problem is detailed in the KEEL repository [1].

The proposed algorithm is composed of a set of different parameters designed to regulate its behaviour:

- The two parameters of the function H: the c exponent and the K, that regulates when to speed up or slow down the simulation.
- Base anomaly threshold in each individual model (before the decision making phase), for both cluster size P, and time alive PT. So clusters with size lesser then P that lived more than PT will be declared as anomalies.
- The aggregation function used to fuse the attractive forces.
- The number of models to use in the decision making scheme.

A small summary of the numerical parameters can be found in Table 2. We can easily fixate some of them if we do not expect significant changes in the outcome $P = 5\%$ of the total mass, and $PT = 10\%$ of the total execution time. These heuristically chosen values work well with most configurations.

In the following, we have explored the effect of the rest of the parameters. For each of the subsequent tasks, the evaluation process is the same: we split the data into $70\%/30\%$ train/test partitions, so that we generate the clusters in the training set, and then classify the instances in the test set.

We have compared the performance of different aggregations for the force vectors and different versions of the H function. Table 3 shows the results using the standard product of the masses instead of the H-function. We found that in this case the summation performed better than the rest of the aggregations. Table 4 shows the results using an H function with static B_1 B_2, where we found that in three cases the summation was the best performant aggregation, the arithmetic mean and the median in one

Table 1: Imabalance ratio, features and samples for each dataset studied.

Dataset	Imb. Ratio	Features	Samples
Ecoli1	3.36	7	336
Ecoli3	8.6	7	336
Glass6	6.38	9	214
Wisconsin	1.86	8	683
Yeast3	8.1	8	1484

Table 2: Brief description of the parameters of the proposed algorithm.

Parameter	Description
c	Exponent present in the H function in Eq. (12).
Static or dynamic	If static, B_1 and B_2 are fixed from the start. If dynamic, B_1 and B_2 change during the execution of the algorithm.
K	Parameter that establishes when the B_1 and B_2 parameters grow, based on the percentage of the total mass of the simulation that is present in a single particle.
P	Maximum anomaly cluster size.
PT	Maximum alive time for an anomaly cluster.
Number of Models	Number of models to use in the decision making phase.
Aggregation function	Function used to aggregate the force vectors for each particle (Summation in the original Gravitational Clustering).

Table 3: F1 score for the proposed algorithm using different aggregation functions for the attraction forces, using the standard product of the masses.

Dataset	Aggregation				
	Sum	FG-Sugeno	Choquet	Arithmetic Mean	Median
Ecoli1	**0.8468**	0.8270	0.8350	0.8331	0.8177
Ecoli3	0.9285	0.9343	**0.9432**	0.9432	0.9154
Glass6	**0.9254**	0.8945	0.8960	0.9057	0.8878
Wisconsin	**0.8062**	0.7971	0.7954	0.7954	0.8006
Yeast3	**0.9347**	0.9239	0.9324	0.9324	0.9163

case each one. Finally, Table 5 shows the results using the dynamic version of the B_1 and B_2 parameters, where we found that the arithmetic mean and the summation each won in two datasets, and the FG-Sugeno in one of them. In general, we found that the summation is a good performing aggregation, compared to both our new integrals and classical ones. We also found that in general terms, the dynamic model of our algorithm perform better than the use the standard product of the masses and the static version of the H function.

However, results changed when we consider the decision making scheme proposed in Section 5.3. We tried to use $5, 10, 15$ and 20 number of models to fuse, and an aggregation function to compute a threshold, as displayed in Eq. (26). In order to reduce the number of possibilities, we use the same aggregation function to fuse the distances and to compute the threshold. We found the best results using the FG-Sugeno integral. Table 6 shows the results of this proposal using different numbers of models in the decision making scheme.

8. Comparison With Other Anomaly Detection Algorithms

In this section, we have compared the results obtained with our newly proposed gravitational algorithm with three other classical anomaly detection algorithms:

1. Local Outlier Factor [9] estimates the data density near one point, based on its k nearest neighbours. Points that have substantially lower density than its neighbours are considered as anomalies.
2. One-Class SVM [21] learns a decision function for anomaly detection, in order to maximize the likelihood of the observed data with respect to the rest.
3. Isolation Forest [23] constructs an ensemble of trees from different subsamples of the original dataset, and then, evaluates the number of partitions for each sample required to isolate the sample in the set of trees.

Table 4: F1 score for the proposed algorithm using different aggregation functions for the attraction forces, using a static version of the H function with $B_1 = 0.2$ and $B_2 = 0.8$.

Dataset	Aggregation				
	Sum	FG-Sugeno	Choquet	Arithmetic Mean	Median
Ecoli1	0.8365	0.8318	0.8318	**0.8477**	0.8074
Ecoli3	**0.9382**	0.9327	0.9316	0.9327	0.9050
Glass6	**0.9282**	0.8888	0.8910	0.8841	0.8935
Wisconsin	0.7985	0.7962	0.7972	0.7962	**0.8128**
Yeast3	**0.9390**	0.9101	0.9101	0.9153	0.8803

Table 5: F1 score for the proposed algorithm using different aggregation functions for the attraction forces, using the dynamic version of the H function.

Dataset	Aggregation				
	Sum	FG-Sugeno	Choquet	Arithmetic Mean	Median
Ecoli1	0.8416	0.8250	0.8407	**0.8458**	0.8157
Ecoli3	**0.9477**	0.8843	0.9419	0.9419	0.8772
Glass6	0.9102	**0.9270**	0.9091	0.9257	0.8933
Wisconsin	**0.8140**	0.7923	0.7884	0.7884	0.7993
Yeast3	0.9306	0.9164	0.9329	**0.9334**	0.7993

Table 6: F1 score for the proposed algorithm using different numbers of models for the decision making phase using the best possible aggregation function (the minimum) to take de decision. We use the FG-Sugeno aggregation function for the attraction forces and using the dynamic version of the H function.

Datasets	Number of Models			
	5	10	15	20
Ecoli1	0.8841	0.8900	0.8947	**0.8952**
Ecoli3	0.8887	0.8918	0.8943	**0.8949**
Glass6	0.8925	0.8925	0.8938	**0.8944**
Wisconsin	0.8767	0.8904	0.8956	**0.8968**
Yeast3	0.8884	0.8897	0.8916	**0.8925**

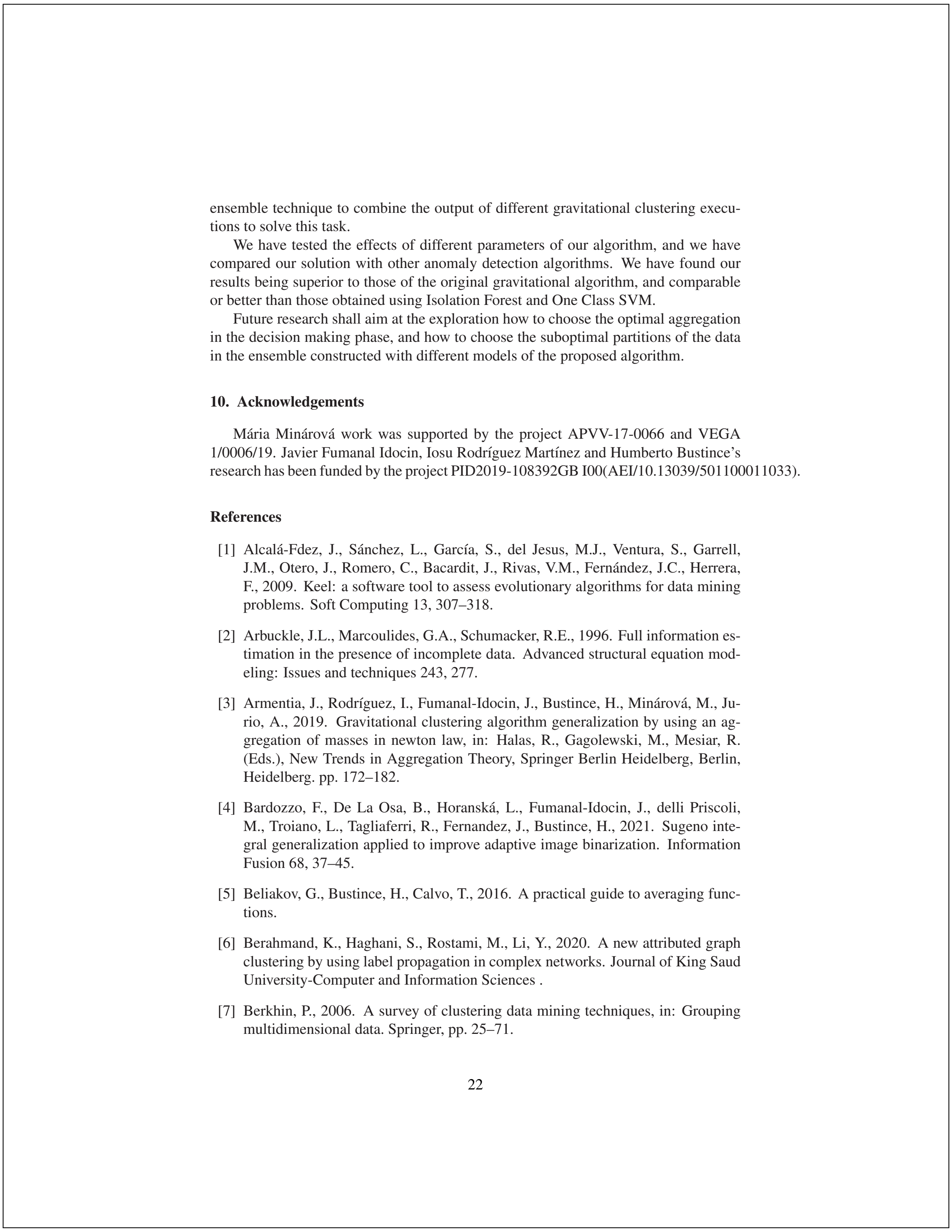
Table 7 shows the results for the different datasets studied in our experimentation for the methods described, and for the original gravitational clustering algorithm. We found our results to be superior to those obtained by other methods in three cases

9. Conclusions

In this work we have presented a new version of the Gravitational clustering algorithm designed to detect anomalies. We have done so by proposing a generalization of the product of the masses using a new class of symmetric and increasing functions that can change during the execution of the algorithm; and a generalization of the summation of forces for a more general aggregation function. Finally, we have proposed an

Table 7: Comparison for different anomaly detection algorithms in five different real-world datasets, using the F1-score as performance metric.

	Ecoli1	Ecoli3	Glass	Wisconsin	Yeast
LOF	0.84029	0.9085	0.8490	0.6881	0.9170
OCSVM	0.6237	0.6303	0.6957	0.8584	0.6676
Isolation Forest	0.8206	0.8748	0.9274	**0.9631**	0.9147
Anomaly-Grav	**0.8634**	**0.9466**	**0.9377**	0.7804	**0.9423**



ensemble technique to combine the output of different gravitational clustering executions to solve this task.

We have tested the effects of different parameters of our algorithm, and we have compared our solution with other anomaly detection algorithms. We have found our results being superior to those of the original gravitational algorithm, and comparable or better than those obtained using Isolation Forest and One Class SVM.

Future research shall aim at the exploration how to choose the optimal aggregation in the decision making phase, and how to choose the suboptimal partitions of the data in the ensemble constructed with different models of the proposed algorithm.

10. Acknowledgements

Mária Minárová work was supported by the project APVV-17-0066 and VEGA 1/0006/19. Javier Fumanal Idocin, Iosu Rodríguez Martínez and Humberto Bustince's research has been funded by the project PID2019-108392GB I00(AEI/10.13039/501100011033).

References

[1] Alcalá-Fdez, J., Sánchez, L., García, S., del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V.M., Fernández, J.C., Herrera, F., 2009. Keel: a software tool to assess evolutionary algorithms for data mining problems. Soft Computing 13, 307–318.

[2] Arbuckle, J.L., Marcoulides, G.A., Schumacker, R.E., 1996. Full information estimation in the presence of incomplete data. Advanced structural equation modeling: Issues and techniques 243, 277.

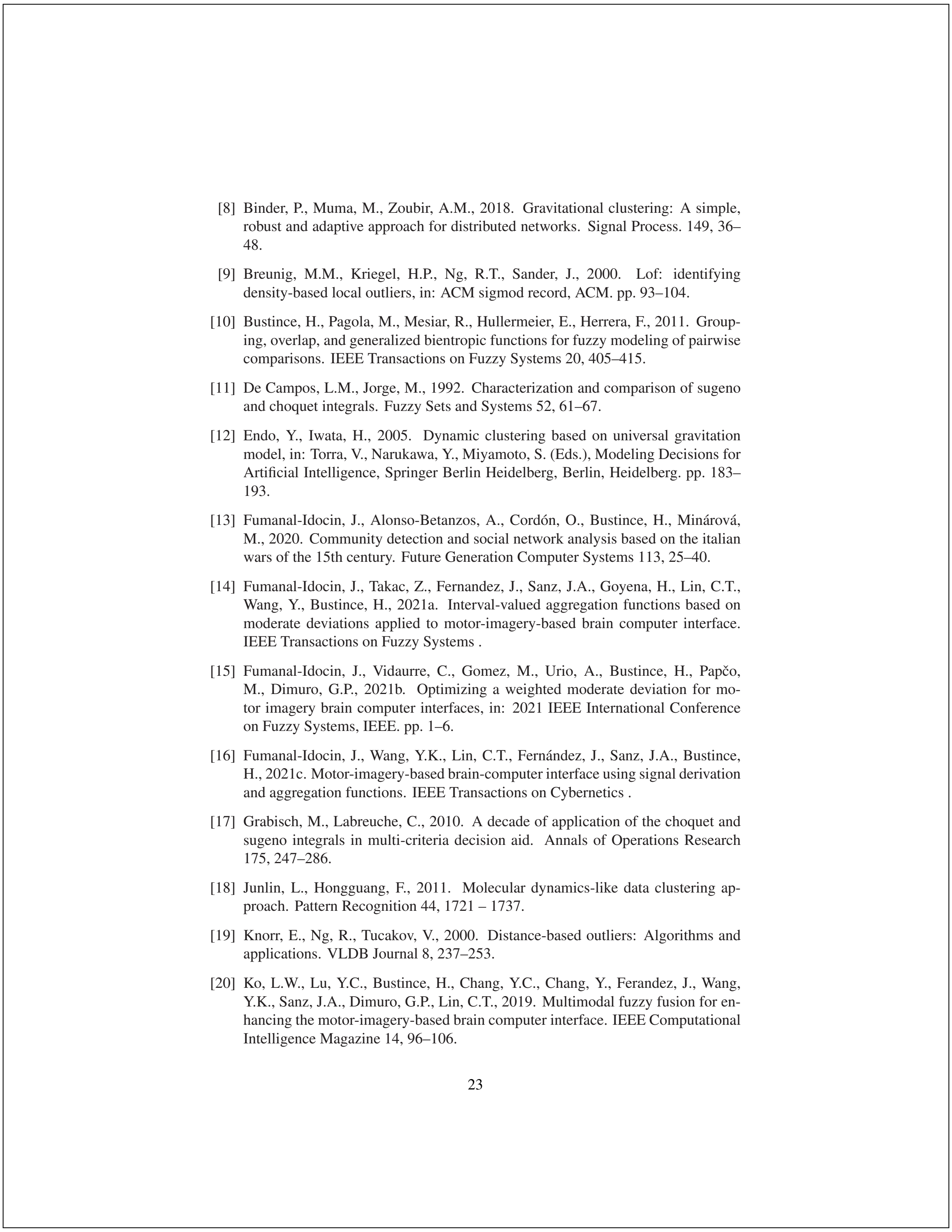
[3] Armentia, J., Rodríguez, I., Fumanal-Idocin, J., Bustince, H., Minárová, M., Jurio, A., 2019. Gravitational clustering algorithm generalization by using an aggregation of masses in newton law, in: Halas, R., Gagolewski, M., Mesiar, R. (Eds.), New Trends in Aggregation Theory, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 172–182.

[4] Bardozzo, F., De La Osa, B., Horanská, L., Fumanal-Idocin, J., delli Priscoli, M., Troiano, L., Tagliaferri, R., Fernandez, J., Bustince, H., 2021. Sugeno integral generalization applied to improve adaptive image binarization. Information Fusion 68, 37–45.

[5] Beliakov, G., Bustince, H., Calvo, T., 2016. A practical guide to averaging functions.

[6] Berahmand, K., Haghani, S., Rostami, M., Li, Y., 2020. A new attributed graph clustering by using label propagation in complex networks. Journal of King Saud University-Computer and Information Sciences .

[7] Berkhin, P., 2006. A survey of clustering data mining techniques, in: Grouping multidimensional data. Springer, pp. 25–71.



[8] Binder, P., Muma, M., Zoubir, A.M., 2018. Gravitational clustering: A simple, robust and adaptive approach for distributed networks. Signal Process. 149, 36–48.

[9] Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J., 2000. Lof: identifying density-based local outliers, in: ACM sigmod record, ACM. pp. 93–104.

[10] Bustince, H., Pagola, M., Mesiar, R., Hullermeier, E., Herrera, F., 2011. Grouping, overlap, and generalized bientropic functions for fuzzy modeling of pairwise comparisons. IEEE Transactions on Fuzzy Systems 20, 405–415.

[11] De Campos, L.M., Jorge, M., 1992. Characterization and comparison of sugeno and choquet integrals. Fuzzy Sets and Systems 52, 61–67.

[12] Endo, Y., Iwata, H., 2005. Dynamic clustering based on universal gravitation model, in: Torra, V., Narukawa, Y., Miyamoto, S. (Eds.), Modeling Decisions for Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 183–193.

[13] Fumanal-Idocin, J., Alonso-Betanzos, A., Cordón, O., Bustince, H., Minárová, M., 2020. Community detection and social network analysis based on the italian wars of the 15th century. Future Generation Computer Systems 113, 25–40.

[14] Fumanal-Idocin, J., Takac, Z., Fernandez, J., Sanz, J.A., Goyena, H., Lin, C.T., Wang, Y., Bustince, H., 2021a. Interval-valued aggregation functions based on moderate deviations applied to motor-imagery-based brain computer interface. IEEE Transactions on Fuzzy Systems .

[15] Fumanal-Idocin, J., Vidaurre, C., Gomez, M., Urio, A., Bustince, H., Papčo, M., Dimuro, G.P., 2021b. Optimizing a weighted moderate deviation for motor imagery brain computer interfaces, in: 2021 IEEE International Conference on Fuzzy Systems, IEEE. pp. 1–6.

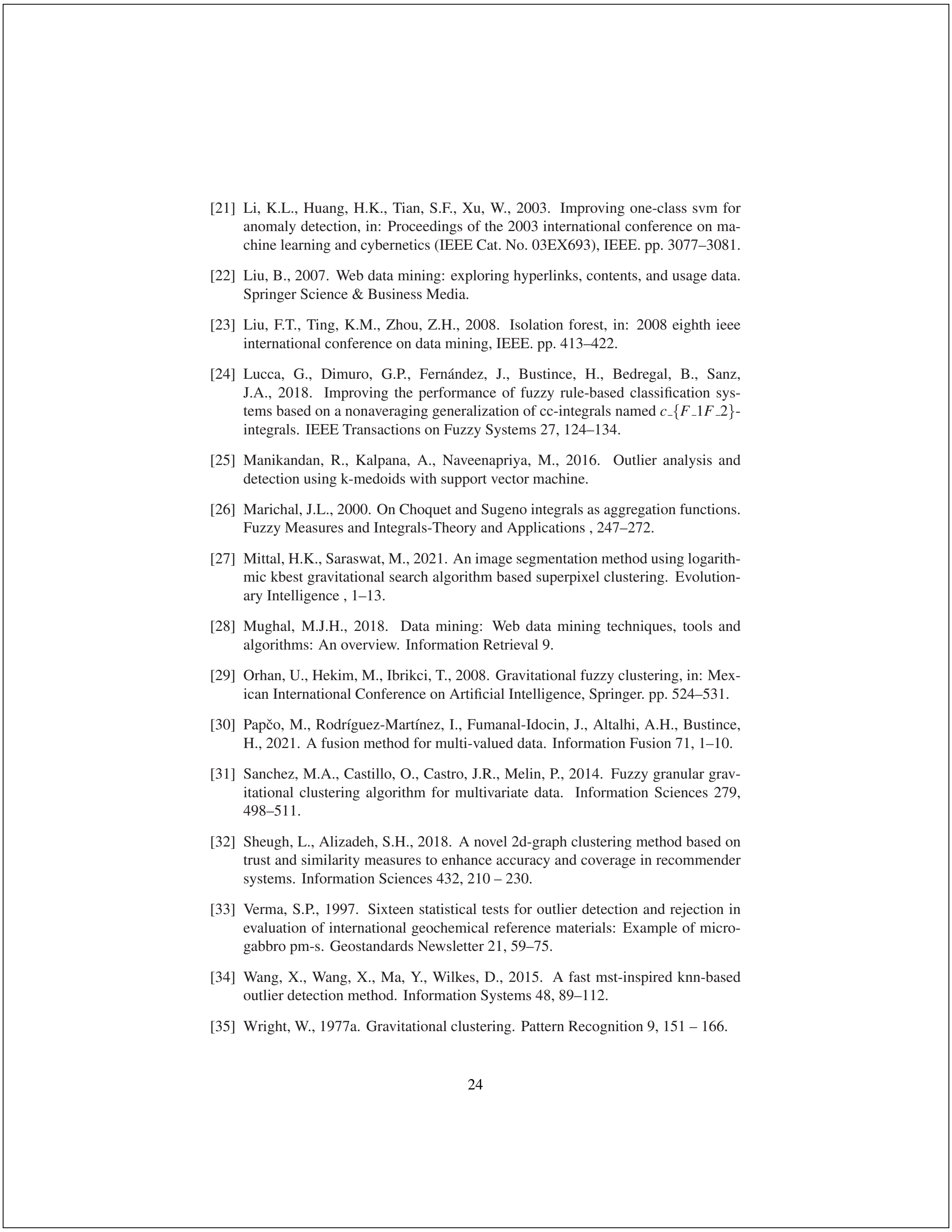
[16] Fumanal-Idocin, J., Wang, Y.K., Lin, C.T., Fernández, J., Sanz, J.A., Bustince, H., 2021c. Motor-imagery-based brain-computer interface using signal derivation and aggregation functions. IEEE Transactions on Cybernetics .

[17] Grabisch, M., Labreuche, C., 2010. A decade of application of the choquet and sugeno integrals in multi-criteria decision aid. Annals of Operations Research 175, 247–286.

[18] Junlin, L., Hongguang, F., 2011. Molecular dynamics-like data clustering approach. Pattern Recognition 44, 1721 – 1737.

[19] Knorr, E., Ng, R., Tucakov, V., 2000. Distance-based outliers: Algorithms and applications. VLDB Journal 8, 237–253.

[20] Ko, L.W., Lu, Y.C., Bustince, H., Chang, Y.C., Chang, Y., Ferandez, J., Wang, Y.K., Sanz, J.A., Dimuro, G.P., Lin, C.T., 2019. Multimodal fuzzy fusion for enhancing the motor-imagery-based brain computer interface. IEEE Computational Intelligence Magazine 14, 96–106.



[21] Li, K.L., Huang, H.K., Tian, S.F., Xu, W., 2003. Improving one-class svm for anomaly detection, in: Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693), IEEE. pp. 3077–3081.

[22] Liu, B., 2007. Web data mining: exploring hyperlinks, contents, and usage data. Springer Science & Business Media.

[23] Liu, F.T., Ting, K.M., Zhou, Z.H., 2008. Isolation forest, in: 2008 eighth ieee international conference on data mining, IEEE. pp. 413–422.

[24] Lucca, G., Dimuro, G.P., Fernández, J., Bustince, H., Bedregal, B., Sanz, J.A., 2018. Improving the performance of fuzzy rule-based classification systems based on a nonaveraging generalization of cc-integrals named $c\_\{F\_1F\_2\}$-integrals. IEEE Transactions on Fuzzy Systems 27, 124–134.

[25] Manikandan, R., Kalpana, A., Naveenapriya, M., 2016. Outlier analysis and detection using k-medoids with support vector machine.

[26] Marichal, J.L., 2000. On Choquet and Sugeno integrals as aggregation functions. Fuzzy Measures and Integrals-Theory and Applications , 247–272.

[27] Mittal, H.K., Saraswat, M., 2021. An image segmentation method using logarithmic kbest gravitational search algorithm based superpixel clustering. Evolutionary Intelligence , 1–13.

[28] Mughal, M.J.H., 2018. Data mining: Web data mining techniques, tools and algorithms: An overview. Information Retrieval 9.

[29] Orhan, U., Hekim, M., Ibrikci, T., 2008. Gravitational fuzzy clustering, in: Mexican International Conference on Artificial Intelligence, Springer. pp. 524–531.

[30] Papčo, M., Rodríguez-Martínez, I., Fumanal-Idocin, J., Altalhi, A.H., Bustince, H., 2021. A fusion method for multi-valued data. Information Fusion 71, 1–10.

[31] Sanchez, M.A., Castillo, O., Castro, J.R., Melin, P., 2014. Fuzzy granular gravitational clustering algorithm for multivariate data. Information Sciences 279, 498–511.

[32] Sheugh, L., Alizadeh, S.H., 2018. A novel 2d-graph clustering method based on trust and similarity measures to enhance accuracy and coverage in recommender systems. Information Sciences 432, 210 – 230.

[33] Verma, S.P., 1997. Sixteen statistical tests for outlier detection and rejection in evaluation of international geochemical reference materials: Example of microgabbro pm-s. Geostandards Newsletter 21, 59–75.

[34] Wang, X., Wang, X., Ma, Y., Wilkes, D., 2015. A fast mst-inspired knn-based outlier detection method. Information Systems 48, 89–112.

[35] Wright, W., 1977a. Gravitational clustering. Pattern Recognition 9, 151 – 166.

[36] Wright, W.E., 1977b. Gravitational clustering. Pattern recognition 9, 151–166.

[37] Xu, R., Wunsch, D., 2008. Clustering. volume 10. John Wiley & Sons.

[38] Yager, R.R., Kacprzyk, J., Beliakov, G., 2011. Recent developments in the ordered weighted averaging operators: theory and practice. volume 265. Springer.

[39] Yang, J., Grunsky, E., Cheng, Q., 2019. A novel hierarchical clustering analysis method based on kullback–leibler divergence and application on dalaimiao geochemical exploration data. Computers & Geosciences 123, 10 – 19.

6.7 Community Detection and Social Network Analysis based on the Italian Wars of the 15th Century

Associated publication:

- Fumanal-Idocin, J., Alonso-Betanzos, A., Cordón, O., Bustince, H., & Minárová, M. (2020). Community detection and social network analysis based on the Italian wars of the 15th century. *Future Generation Computer Systems, 113, 25-40.*

  Status: Published.

  Impact Factor (JCR 2021): 7.307.

  Categories:

  Computer Science, Theory & Methods. Ranking 10/109.

Future Generation Computer Systems 113 (2020) 25–40

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

Community detection and social network analysis based on the Italian wars of the 15th century

J. Fumanal-Idocin [a], A. Alonso-Betanzos [d], O. Cordón [e], H. Bustince [a,b,*], M. Minárová [c]

[a] Public University of Navarra and Institute of Smart Cities, Campus Arrosadia s/n, 31006 Pamplona, Spain
[b] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[c] Slovak University of Technology in Bratislava, Radlinského 11, 81005 Bratislava, Slovakia
[d] CITIC. University of A Coruña, 15071 A Coruña., Spain
[e] Andalusian Research Institute DaSCI, "Data Science and Computational Intelligence", University of Granada, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 8 February 2020
Received in revised form 28 April 2020
Accepted 19 June 2020
Available online 3 July 2020

Keywords:
Social network
Community detection
Human social behaviour
Simulation
Multi-agent systems

ABSTRACT

In this contribution we study social network modelling by using human interaction as a basis. To do so, we propose a new set of functions, affinities, designed to capture the nature of the local interactions among each pair of actors in a network. By using these functions, we develop a new community detection algorithm, the Borgia Clustering, where communities naturally arise from the multi-agent interaction in the network. We also discuss the effects of size and scale for communities regarding this case, as well as how we cope with the additional complexity present when big communities arise. Finally, we compare our community detection solution with other representative algorithms, finding favourable results.

1. Introduction

Network analysis has become an important tool to study systems composed of interacting agents, such as proteins or human societies [1–5]. One of the key ideas in social sciences is the one that we, human beings, are embedded by our own social nature in a complex web of social relations and interactions. Traditionally, this hierarchy that we have formed has been modelled as a network, where each person is represented as a node that is connected to others according to some criteria. Social networks analysis stands as an appropriate tool to understand many characteristics of the human behaviour, as it seems that many of us are deeply affected by the social structure in which we take part [6]. Adjacency matrices are the most common form of network representation [7]. However, if there is more data available to construct the network, a more complex model can be used [8]. Depending on the context, different models have been developed to mimic human behaviour: in an educational organization [9] and when dealing with conversations [10].

There are many problems related to social networks, such as information diffusion [11], social circles detection [12], coalition formation [13], recommendation systems [14] or user behaviour prediction [15]. Social networks can also be seen as multi-agent systems, and it is possible to study their emergent properties [16]. One key step to analyse a network is to identify its community structure: the groups of nodes that can be identified as a functional sub-partition of the graph [17,18] e.g a group of friends or a protein complex [19]. Communities are important because we can infer significant knowledge from a node or a set of nodes if we know whether or not they share the same community and what kind of community it is. One classical method to develop community detection is hierarchical clustering [20], although there are many algorithms performing community detection in a social network that improve the results obtained by this method. Authors in [21] use a genetic algorithm to identify densely connected groups of nodes. The algorithm in [22] performs an initial community detection in the most important nodes in the network, and then labels the rest of them.

Modularity is a measure that quantifies the quality of a graph partition into different modules. Networks with high modularity have a high number of edges between the nodes within modules, and a low number of them between nodes in different modules [23]. There are many modularity-based methods to perform community detection. The proposal in [24] uses the idea of modularity and it is much faster than previous algorithms. Authors in [25] propose a modularity optimization approach to work on large networks, with this being one of the most extensively used community detection methods.

[*] Corresponding author at: Public University of Navarra and Institute of Smart Cities, Campus Arrosadia s/n, 31006 Pamplona, Spain.
E-mail addresses: javier.fumanal@unavarra.es (J. Fumanal-Idocin), amparo.alonso.betanzos@udc.es (A. Alonso-Betanzos), ocordon@decsai.ugr.es (O. Cordón), bustince@unavarra.es (H. Bustince), maria.minarova@stuba.sk (M. Minárová).

https://doi.org/10.1016/j.future.2020.06.030
0167-739X/© 2020 Elsevier B.V. All rights reserved.

In this paper, our goal is to solve some of the social network analysis problems that we have identified both in actor interaction and community detection. First, current community detection algorithms do not take into account neither the different nature of the involved human beings when applied to social networks, nor the impact to its structure. We have also noted how scale in social networks can alter its dynamics. Besides, when it comes to community detection, many of the existing algorithms have problems in densely connected graphs and the appropriate algorithms are often awfully time consuming [26].

To solve these problems, we studied how human-inspired algorithms can lead to a better understanding of the structure of a social network and its communities. In contrast to existing literature, which uses an adjacency matrix to model a network [27], we propose a set of functions for the sake of better capturing the relationships between each pair of actors in the social network. We model an algorithm that forms communities in a similar fashion to real social networks. Finally, we propose a new representation space for these graphs, called "affinities", that can be calculated from the original adjacency graph. These affinities model how strong the relationship between two nodes according to different criteria is. Based on these functions, we have developed a new algorithm based on the gravitational algorithm described in [28]. We have called our new algorithm to perform community detection the Borgia Clustering.

Using the affinity functions and the Borgia Clustering, we aim at having a better understanding of local interactions and how they can affect global dynamics. Regarding community detection, our target is to obtain a state-of-the-art algorithm to perform this task. Such algorithm would be able to generate a dendrogram faithfully reflecting the evolution of the network in the clustering process and choose the right configuration inside it.

To test the quality of our proposals, we have studied how the new affinities affect the interaction between actors in some real-world datasets. We have also tested our community detection method in four different networks.

The rest of the paper is as follows. In Section 2 we explain the basics of the gravitational algorithm and graph theory and we explain what is a T-Norm and an Overlap. In Section 3 we explain the new representation space for social networks based on the traditional adjacency matrix used for graph representation. In Section 4 we introduce the Borgia Clustering algorithm, the historical moment that inspired it, and how it did so. In Section 5 we discuss some issues that we found working with communities of different sizes. In Section 6 we test our algorithm on three real world datasets and in Section 7 we compare the quality of our solution against other representative algorithms. Finally, in Section 8 we summarize the whole work and state some future guidelines.

2. Preliminaries

In this section we will briefly explain some of the already-existing concepts related to some of the new proposals in this paper:

- The gravitational clustering algorithm.
- Aggregation functions.
- Graph theory.

2.1. Algorithm of gravitational clustering

The algorithm of Gravitational Clustering [28] employs the Newton gravitational law within the process of clustering. In this algorithm, each observation is a particle that attracts the others according to their distances and masses. When two particles are closer than the collision distance, they are merged into a single one. Its mass is the sum of those particles and its position is their centre of masses. This process repeats until only one particle exists. This algorithm results in a dendrogram containing each particle fusion. Finally the most stable configuration (with the largest lifespan) is taken as the resulting one. The scheme is as follows:

We suppose that we have n particles $p_1, \ldots, p_n$, with their positions $s_1, \ldots, s_n \in R^n$. We also have two parameters: ϵ, which establishes the collision distance for two particles, and δ which determines the movement for the fastest particle in each iteration.

1. Initially we assign a mass (m_i) 1 to each particle p_i.
2. We fix real positive parameters ϵ and δ
    - We utilize δ for determining the actual time step longitude, dt. It is the time in which the fastest particle moves.
    - If in a moment two particles find themselves in a distance less than ϵ we unify them in one particle. The mass of the new resulting particle is the sum of both masses and its position is their centre of masses. Likewise in the case of three or more particles.
3. Initial time is set to $t = 0$.
4. We repeat the following steps (i)-(iv) until a single particle remains.
    (i) In each time interval $[t, t+dt]$, for each particle p_i we compute its movement influencing function:

$$g(i, t, dt) = \frac{1}{2} G \sum_{j \neq i} \frac{m_i(t) m_j(t)}{m_i(t)} \frac{s_j(t) - s_i(t)}{|s_j(t) - s_i(t)|^3} dt^2 \qquad (1)$$

    where G is a positive constant.
    (ii) For each particle i, its new position is:

$$s_i(t + dt) = s_i(t) + g(i, t, dt)$$

    (iii) We increment t to $t + dt$.
    (iv) If two particles i and j are closer than ϵ, they are fused as explained above.

After the algorithm ends, we have just one particle. The duration of the entire process is denoted by T. We measured the duration of each iteration, as well, with the aim of being able to detect afterwards the most stable configuration. The final result is the configuration that lasted the longest period of the simulated time.

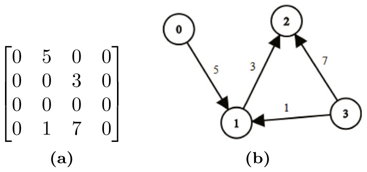
2.2. Aggregation functions

Definition 2.1. Aggregation functions are used to combine the information of multiple numerical sources into a single one. A function $A : [0, 1]^n \rightarrow [0, 1]$ is said to be n-ary aggregation function is said to be a n-ary aggregation function if the following conditions hold [29]:

- A is increasing in each argument: $\forall i \in \{1, \ldots, n\}, i < y, A(x_1, \ldots, i, \ldots, x_n) \leq A(x_1, \ldots, y, \ldots, x_n)$
- $A(0, \ldots, 0) = 0$
- $A(1, \ldots, 1) = 1$

Definition 2.2. A bivariate aggregation function T: $[0, 1]^2 \rightarrow [0, 1]$ is a t-norm if, $\forall x, y, z \in [0, 1]$, it satisfies the following properties:

- Commutativity: $T(x, y) = T(y, x)$
- Associativity: $T(x, T(y, z)) = T(T(x, y), z)$.
- Boundary condition: $T(x, 1) = x$



$\begin{bmatrix} 0 & 5 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 7 & 0 \end{bmatrix}$
(a)
0
1
2
3
5
3
7
1
(b)

Fig. 1. Directed and weighted graph. a. Adjacency matrix of the graph. **b**. Visual representation of the graph.

Two prototypical examples of T-norms are the minimum and the product.

2.3. Graphs

A graph G is represented as $G(V, E)$ where V is a set of vertices and E is a set of edges that connect some pairs of vertices in the graph G.

There are different kinds of graphs, depending on the information related to each edge. In case we have some information regarding the strength of the relationships, we call it a weighted graph. If we do not have such information, it is called an unweighted graph. Edges can also be undirected, when an edge between $V_i \leftrightarrow V_j$ represents a bidirectional relationship, or directed, when the edge between $V_i \rightarrow V_j$ can be different from the edge $V_j \rightarrow V_i$.

Graphs can be characterized by using many statistics such as the average number of connections per node, the average path lengths between nodes, etc.

Graphs can be modelled by using different representations. The most common ones are the adjacency or connectivity matrix and the adjacency list.

The adjacency matrix A of a graph G is a $N \times N$ matrix where N is the number of nodes in G. Each entry A_{ij} in this matrix corresponds to the value associated with the $V_i \rightarrow V_j$ edge. If the graph is unweighed, those values will be 0 or 1, while if the graph is weighted, those values will be the corresponding weights for each edge (Fig. 1). The adjacency list is similar to the adjacency matrix, but instead of storing $N \times N$ elements, we store a list for each vertex containing the rest of nodes which are communicated with it.

3. Affinity functions as an actor representation

We define "Affinities" as a set of functions of two different actors, Actor$_x$ and Actor$_y$, establishing their mutual relation using C:

$A_C : [Actor_x, Actor_y] \rightarrow [0, 1]$

Usually, this C is the adjacency matrix that quantifies the relationships in each pair of actors, although C can be, for example, another affinity matrix, or a list of them. The affinity between two actors shows how strongly these two are connected. Since affinities are not necessarily symmetrical, the strength of this interaction depends on who the sender and receiver are, as happens in human interaction e.g. unrequited love.

We proceed to list some affinity functions:

- **Best Friend affinity (BF)**: the affinity of the actor $Actor_x$ over the $Actor_y$ is defined as the percentage of the total connectivity of Y that corresponds to $C_{x,y}$.
- **Best Common Friend affinity (BCF)**: the affinity between two actors is defined as the biggest affinity common to the both of them. It can be computed using both the adjacency matrix or another previously calculated affinity.

Table 1
Formula proposed for each of the affinities. C is the adjacency matrix and N is the total number of actors in C.

Affinity$_{x,y}$	*Formula*
Best friend	$A_C(x,y) = \frac{C_{x,y}}{\sum_{a=1}^{N} C_{x,a}}$
Best Common friend	$A_C(x,y) = Max\{Min(C_{x,z}, C_{y,z})\}/\sum_{a=1}^{N} C_{x,a}$
Friends forever	$A_C(x,y) = \sum(\frac{C_{x,y}(t)}{\sum_{a=1}^{N} C_{x,a}(t)})\frac{1}{\lvert T\rvert}, \ \forall t \in T$
Social networking	$A_C(x,y) = Mean(A'_C(x',y)) \ \forall x'$, such that $A'_C(x,x') > 0$
Machiavelli	$A_C(x,y) = 1 - \frac{abs(I_x - I_y)}{Max(I_x, I_y)}, I_a = Sum(Degree(x')) \ \forall x$ such that $C(a,x') > 0$

- **Friends Forever affinity (FF)**: the affinity of two actors reflects the durability of the relation in time.
- **Social Networking affinity (SN)**: the affinity between two actors, Actor$_x$ and Actor$_y$, is based on the affinities of the actors connected to Actor$_x$ with respect to Actor$_y$.
- **Machiavelli Affinity (MA)**: the affinity between two actors is based on the social structure that is built around the two of them.

All but the Machiavelli affinity are personal affinities. A formula to calculate each affinity function is in Table 1.

There are, mainly, two types of affinity functions: personal affinities and structural affinities. Personal affinities establish the strength of a interpersonal connection $Actor_x \rightarrow Actor_y$ using their respective connections and shared friends. Structural affinities quantify the relationship of a pair of actors based on the properties of their nodes, such as their degree or betweenness.

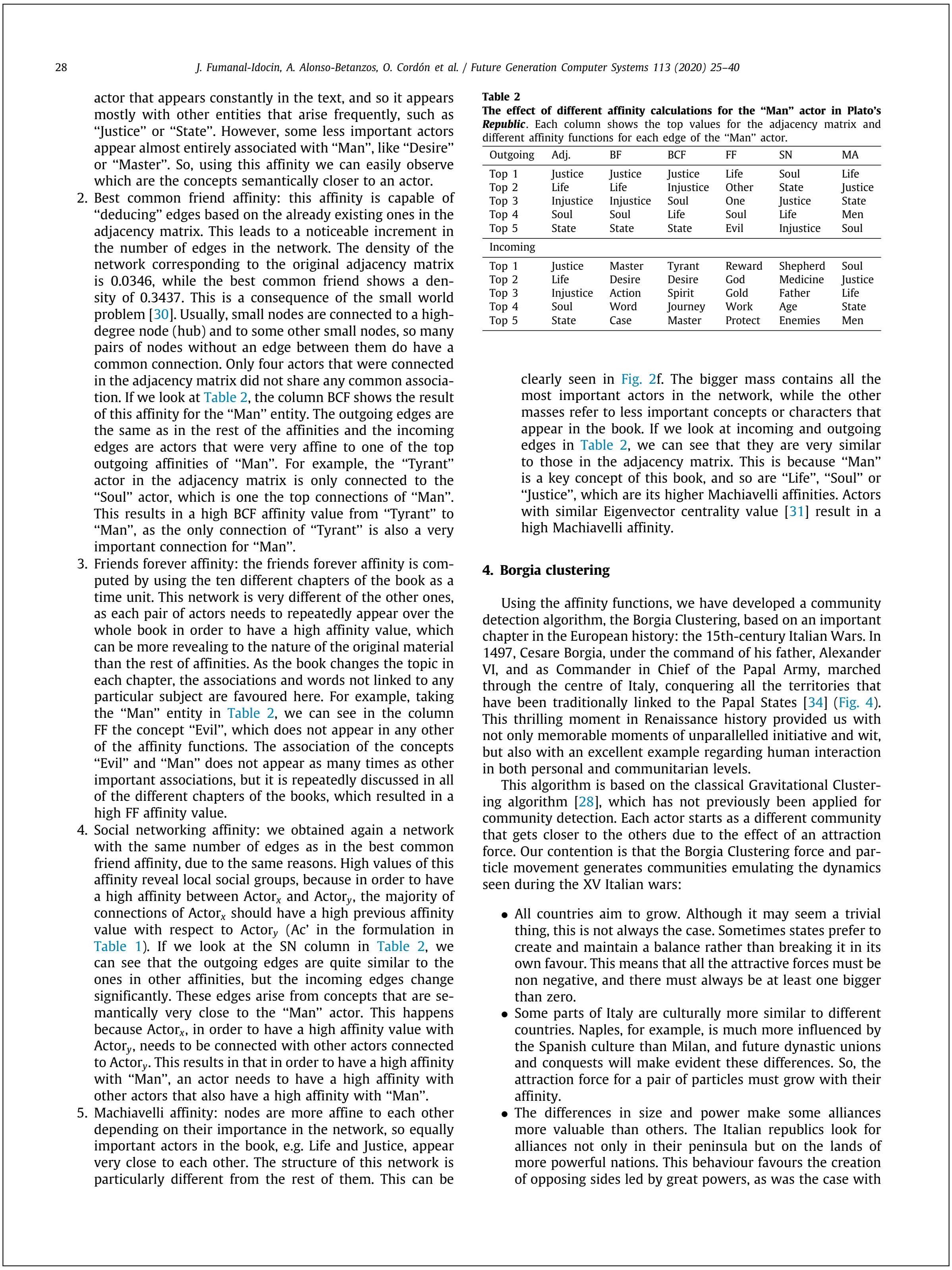
One particular difference between these two types is that the expected value of a personal affinity is affected by degree of an actor, but this does not necessarily happen in structural affinities. This is because the personal affinity functions behave like a zero-sum game. It can be easily seen in the best friend affinity, where the higher the number of connections the higher the denominator value in the expression.

3.1. Effects of different affinity functions in plato's Republic

For illustration, we consider the network of word association in classical literature. In particular, we take *Republic*, by Plato. We visualize the resulting network (Fig. 2) and the heatmap for the original network (Fig. 3) for each different affinity. In these networks, each node corresponds to a different word in the original work, and its size is directly proportional to its degree. In the co-occurrences network (Fig. 3a), each edge represents the number of times when two words appear together in a paragraph, and in the affinity graphs (Fig. 3b–f), each edge is the affinity value for each pair of nodes. For the sake of clarity, only the 130 most frequent words are present in each network.

We have computed all the different affinities provided in Table 1. Depending on the affinity function used, the resulting edges and their weights can be very different. In Table 2 we have computed the five affinities for the entity "Man", and we proceed to discuss the results obtained:

1. Best friend affinity: we obtained a network with the same edges as the original adjacency matrix, but with a weight for each edge. For example, let us consider the "Man" actor entity. In Table 2 there are the top incoming and outgoing BF affinities for the "Man" entity. Outgoing entities are the same as in the adjacency matrix. However, the incoming edges are terms that associate exclusively or almost exclusively with this actor. This happens because "Man" is an



actor that appears constantly in the text, and so it appears mostly with other entities that arise frequently, such as "Justice" or "State". However, some less important actors appear almost entirely associated with "Man", like "Desire" or "Master". So, using this affinity we can easily observe which are the concepts semantically closer to an actor.

2. Best common friend affinity: this affinity is capable of "deducing" edges based on the already existing ones in the adjacency matrix. This leads to a noticeable increment in the number of edges in the network. The density of the network corresponding to the original adjacency matrix is 0.0346, while the best common friend shows a density of 0.3437. This is a consequence of the small world problem [30]. Usually, small nodes are connected to a high-degree node (hub) and to some other small nodes, so many pairs of nodes without an edge between them do have a common connection. Only four actors that were connected in the adjacency matrix did not share any common association. If we look at Table 2, the column BCF shows the result of this affinity for the "Man" entity. The outgoing edges are the same as in the rest of the affinities and the incoming edges are actors that were very affine to one of the top outgoing affinities of "Man". For example, the "Tyrant" actor in the adjacency matrix is only connected to the "Soul" actor, which is one the top connections of "Man". This results in a high BCF affinity value from "Tyrant" to "Man", as the only connection of "Tyrant" is also a very important connection for "Man".
3. Friends forever affinity: the friends forever affinity is computed by using the ten different chapters of the book as a time unit. This network is very different of the other ones, as each pair of actors needs to repeatedly appear over the whole book in order to have a high affinity value, which can be more revealing to the nature of the original material than the rest of affinities. As the book changes the topic in each chapter, the associations and words not linked to any particular subject are favoured here. For example, taking the "Man" entity in Table 2, we can see in the column FF the concept "Evil", which does not appear in any other of the affinity functions. The association of the concepts "Evil" and "Man" does not appear as many times as other important associations, but it is repeatedly discussed in all of the different chapters of the books, which resulted in a high FF affinity value.
4. Social networking affinity: we obtained again a network with the same number of edges as in the best common friend affinity, due to the same reasons. High values of this affinity reveal local social groups, because in order to have a high affinity between Actor$_x$ and Actor$_y$, the majority of connections of Actor$_x$ should have a high previous affinity value with respect to Actor$_y$ (Ac' in the formulation in Table 1). If we look at the SN column in Table 2, we can see that the outgoing edges are quite similar to the ones in other affinities, but the incoming edges change significantly. These edges arise from concepts that are semantically very close to the "Man" actor. This happens because Actor$_x$, in order to have a high affinity value with Actor$_y$, needs to be connected with other actors connected to Actor$_y$. This results in that in order to have a high affinity with "Man", an actor needs to have a high affinity with other actors that also have a high affinity with "Man".
5. Machiavelli affinity: nodes are more affine to each other depending on their importance in the network, so equally important actors in the book, e.g. Life and Justice, appear very close to each other. The structure of this network is particularly different from the rest of them. This can be clearly seen in Fig. 2f. The bigger mass contains all the most important actors in the network, while the other masses refer to less important concepts or characters that appear in the book. If we look at incoming and outgoing edges in Table 2, we can see that they are very similar to those in the adjacency matrix. This is because "Man" is a key concept of this book, and so are "Life", "Soul" or "Justice", which are its higher Machiavelli affinities. Actors with similar Eigenvector centrality value [31] result in a high Machiavelli affinity.

Table 2
The effect of different affinity calculations for the "Man" actor in Plato's *Republic*. Each column shows the top values for the adjacency matrix and different affinity functions for each edge of the "Man" actor.

Outgoing	Adj.	BF	BCF	FF	SN	MA
Top 1	Justice	Justice	Justice	Life	Soul	Life
Top 2	Life	Life	Injustice	Other	State	Justice
Top 3	Injustice	Injustice	Soul	One	Justice	State
Top 4	Soul	Soul	Life	Soul	Life	Men
Top 5	State	State	State	Evil	Injustice	Soul
Incoming						
Top 1	Justice	Master	Tyrant	Reward	Shepherd	Soul
Top 2	Life	Desire	Desire	God	Medicine	Justice
Top 3	Injustice	Action	Spirit	Gold	Father	Life
Top 4	Soul	Word	Journey	Work	Age	State
Top 5	State	Case	Master	Protect	Enemies	Men

4. Borgia clustering

Using the affinity functions, we have developed a community detection algorithm, the Borgia Clustering, based on an important chapter in the European history: the 15th-century Italian Wars. In 1497, Cesare Borgia, under the command of his father, Alexander VI, and as Commander in Chief of the Papal Army, marched through the centre of Italy, conquering all the territories that have been traditionally linked to the Papal States [34] (Fig. 4). This thrilling moment in Renaissance history provided us with not only memorable moments of unparallelled initiative and wit, but also with an excellent example regarding human interaction in both personal and communitarian levels.

This algorithm is based on the classical Gravitational Clustering algorithm [28], which has not previously been applied for community detection. Each actor starts as a different community that gets closer to the others due to the effect of an attraction force. Our contention is that the Borgia Clustering force and particle movement generates communities emulating the dynamics seen during the XV Italian wars:

- All countries aim to grow. Although it may seem a trivial thing, this is not always the case. Sometimes states prefer to create and maintain a balance rather than breaking it in its own favour. This means that all the attractive forces must be non negative, and there must always be at least one bigger than zero.
- Some parts of Italy are culturally more similar to different countries. Naples, for example, is much more influenced by the Spanish culture than Milan, and future dynastic unions and conquests will make evident these differences. So, the attraction force for a pair of particles must grow with their affinity.
- The differences in size and power make some alliances more valuable than others. The Italian republics look for alliances not only in their peninsula but on the lands of more powerful nations. This behaviour favours the creation of opposing sides led by great powers, as was the case with

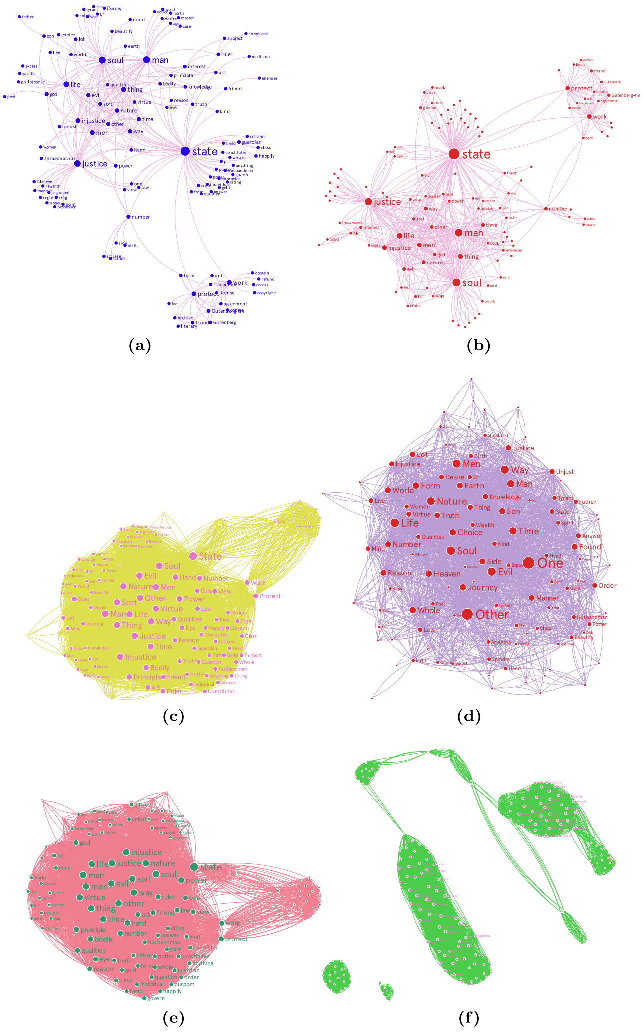


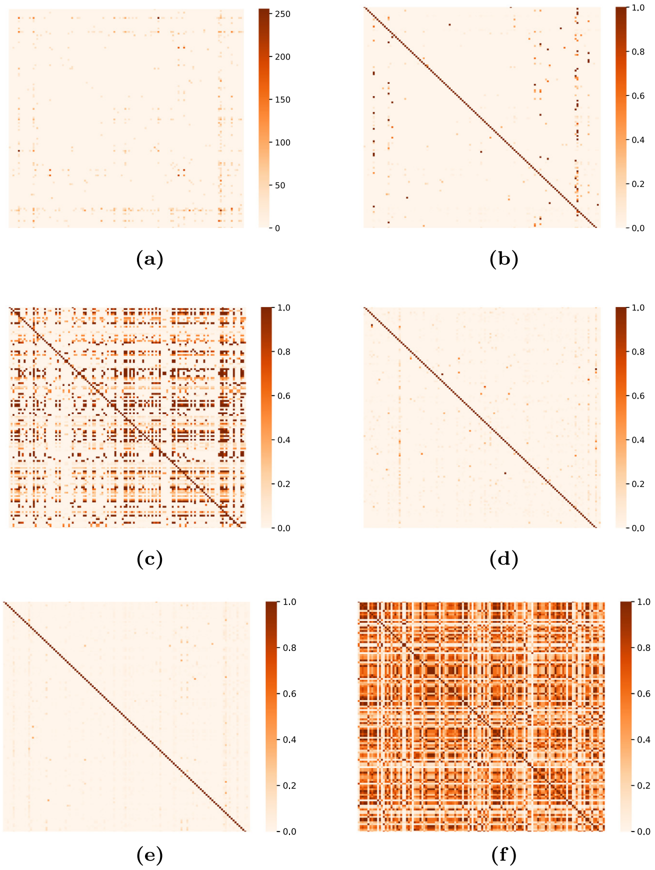
Fig. 2. Network representation of the social network for the Plato's *Republic*. **a**. Co-appearances network. **b**. Best friend affinity network. **c**. Best common friend affinity network. **d**. Friends forever affinity network. **e**. Social network affinity network. **f**. Machiavelli affinity network. All of them have been drawn using the Force Atlas 2 algorithm [32].

the many Franco-Spanish wars that occurred in the fifteenth and sixteenth centuries. The attractive force of an actor must grow with their inherent value and influence in the net.

- There were many countries of different size and importance involved in that historical event. France and Spain were the biggest ones, followed by Milan, Venice, Florence and the Kingdom of Naples. There were also the Papal States and the "independent lords" in central Italy. Contrary to naive thinking, the smallest counties were not conquered by the biggest countries, which are the ones with bigger armies, but by the Papal States. So, even though more size implies more attractive force, this must not be the only factor, and the actors size and scale must be also taken into account.

To represent each actor, we use a combination of the best friend affinity matrix and the best common friend affinity matrix, and an influence matrix, S, based on the former. By using the best friend affinity, we favour strong pair-wise interactions, and with the best common friend affinity, we also favour the formation of communities whose members share a high number of friends. Also, each actor has a "social value", equal to its number of connections, that reflects its popularity.

The actors attract to each other in a simulation of gravitational-like force, depending on their social value, affinity



(a) (b) (c) (d) (e) (f)

Fig. 3. Heatmap representation of the social network in Plato's *Republic*. a The original matrix of co-occurrence of each pair of words. **b** The affinity matrix for the Best friend affinity. **c**. The affinity matrix for the Best Common Friend. **d**. The affinity matrix for the Friend Forever affinity. **e**. The affinity matrix for the Social Networking affinity **f**. The affinity matrix for the Machiavelli affinity.

and distance. We consider that two actors a, b collide when the value of $S(a, b)$ is bigger than $S(b, b)$. Then, we interpret that the actor a has as much influence over the actor b as b has over itself. In that moment they are fused to form a new community. As a result of this, the most affine pairs of actors will naturally join first and start forming communities.

We have decided to construct the Borgia Clustering algorithm by modifying the classical gravitational algorithm. because it can be easily modified to apply our three ideas to effectively model our wanted dynamics and still keep the physical interpretation. To obtain the desired behaviour, we have performed the following modifications:

1. Particles in the original algorithm have been substituted by actors.
2. We have revamped the attraction force in a way that now it takes the size, the distance and social value of each actor into account.
3. The collision condition and fusion procedure for two particles has been adapted to actors.
4. We have replaced the idea of position by the idea of influence. The position matrix has been substituted by an influence matrix, S.
5. Besides the influence matrix, we keep an Affinity matrix, A_C, that contains the affinity for each pair of actors and/or communities alongside the execution of the algorithm.

Finally, we have also studied how to optimize the delta parameter, which controls the maximal movement of each particle for each iteration, to speed up the computation.

4.1. Particle modification

In social networks, each particle is not a point in the space, but an actor. Actors have a set of additional properties, usually related to the semantic information available for each one. The most important one is the connectivity, which allows us to compute the best friend and best common friend affinity.

Actors have an initial social value, m, that initially corresponds to its degree, that is similar to the concept of particle mass in

(a) (b)

Fig. 4. Italy in times of the Borgias. a. Italy's lordships before Cesare Borgia's campaign as Commander in Chief of the Papal Army. Each colour represents a different faction. There is a large amount of independent and small counties in central Italy, more precisely, under *de iure* territories of the Papal States [33]. **b.** A portrait of Cesare Borgia circa 1500. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the original algorithm. This allows us to display two well-known social behaviours: popular people are more socially attractive than people with fewer friends. This fact is called cumulative advantage in network theory [35] (from the physical point of view, a greater mass implies a greater attractive force). Furthermore, people with few friends try to hang out with popular people, but not the reverse (greater mass also implies less movement).

The concept of particle position is substituted for the actor pairwise-influence, denoted as the matrix S, that is built up by using an affinity matrix. The value s_{ij} reflects the influence that the actor j has over the actor i. The self-influences $(s_{i,i})$ are set to 1. Actors will interact with each other along time and they will move closer as time passes due to this interaction. Besides, self-influences will decrease during the execution of the algorithm.

The matrix A_C is initially the same as the matrix S, but A_C only changes its values when two actors collide and form a new community.

4.2. Actor fusion

The condition for checking whether two actors have collided using the Euclidean distance is not good enough in this context due to the curse of dimensionality. Thus, we have used the idea of influence instead of position to check whether two actors should be fused or not.

Each actor starts as being completely influential over itself. During the execution of the algorithm, actors attract to each other, causing a reduction in their own self-influence, and augmenting the influence other actors have over them. When the influence of one actor over another is greater than the self-influence of that actor, those actors will collapse into a new community.

The fusion of two actors a, b results in the emergence of a new actor whose social value is the sum of m_a and m_b, whose position is their centre of masses of s_a and s_b, and whose affinities in the affinity matrix are calculated using formula (2).

$$A_C(ab) = \frac{m_a A_C(a) + m_b A_C(b)}{m_a + m_b} \tag{2}$$

4.3. New attraction formula

We have modified the attraction formula to take into account not only the mass and distance, but the whole set of characteristics present in each actor.

We take into account the bilateral relationships using the A_C and S matrices. The A_C matrix returns the affinity for a pair of actors/communities, and we use the S to compute the Euclidean distances between them, so that the more common or similar affinities between two actors are, the higher the attraction force will be.

We also need to take into account the size of the actor/community. To do so, we add a "greedy expanse" penalization parameter, p. This p penalizes the size of the actor in a non-linear way, so that the bigger it gets, the more difficult to move. The penalization is computed as $\frac{1}{m_x^p}$ for each actor x. This idea used here for the sake of coping with size is what we have called the "Early Roman policy", in Section 5.

The product of the masses and the A_C are aggregated by using a T-norm as aggregation function [36]. Using a T-norm is important for computational speed. T-norms are always less than or equal to the minimum function. If the $A_C(x, y) = 0$, we do not need to calculate the attraction force between x and y because we know that it must be 0.

The computational cost of the original gravitational algorithm is $O(kn^2)$, where k refers to the number of iterations and n to the number of particles. Because of the T-norm aggregation, in our case, the cost is $O(kl)$, l representing the number of edges in the graph. This is a significant result since the maximum possible value for l in a undirected network is n^2 and real networks are usually sparse. Therefore, the complexity order reduction is actually very large.

As a result, the final formula for the attraction force is the following:

$$\mathbf{F}_{xy} = \frac{T((m_x m_y)^c, A_C(x, y))}{m_x^p}, \frac{\mathbf{s}_x - \mathbf{s}_y}{|\mathbf{s}_x - \mathbf{s}_y|^3} dt \tag{3}$$

In it, T stands for a t-norm function, A_C for the chosen affinity function, $\mathbf{s}_x$ for the influence vector of actor x.

4.4. Choosing a configuration

In the original gravitational algorithm, the configuration with the longest life in simulated time is chosen as the final one. This criterion is extremely fast to compute but tends to return a very small number of communities. This happens because in the last steps of the algorithm we only have a reduced number of particles that move very slowly because they are very heavy.

There is no problem if the desired number of communities is low (< 5) but in case we want more, it is necessary to add an extra term to measure the quality, R, of each configuration, Z:

$$R(Z) = SimulatedTime(Z) * log(NumCommunities(Z)) \tag{4}$$

where q is a partition of the graph. By using this formula, we reward both stability in time and a higher number of different communities.

It is also possible to specify the exact number of communities wanted.

4.5. Formulation of the algorithm

To sum up, the formulation of the Borgia Clustering algorithm is the following one:

1. We assign a social value to each actor equal to its degree in the original network.
2. We set t as 0 and the parameter δ. The δ parameter restricts the shift magnitude in each iteration for the fastest particle. Then, the dt of each particular iteration is computed based on the movement of the fastest particle. The rest of particles' movement is computed based on this dt value.
3. We compute the affinity matrix, A_C. Then, we set the initial influence matrix, S, equal to A_C.
4. Next, clustering and attracting steps alternate alongside with the repetition of a) - d) steps until all actors are fused into one.

   (a) The function for driving the movement of each actor i in a time interval $[t, t+dt]$ is:

   $$\mathbf{g}_i(t) = \frac{1}{m_i^p} \sum_{j \neq i} T((m_x m_y)^c, A_C(i,j)) \frac{\mathbf{s}_j(t) - \mathbf{s}_i(t)}{|\mathbf{s}_j(t) - \mathbf{s}_i(t)|^3} dt \tag{5}$$

   (b) The fastest actor is indexed as F

   $F = arg(\max_i\{|\mathbf{g}_i|(t)\})$,

   $dt(t)$ for the next step is computed from $|\mathbf{g}_F(t)| = \delta$:

   $$\delta = \frac{1}{m_F^p} \sum_{j \neq F} T((m_x m_y)^c, A_C(F,j)) | \frac{\mathbf{s}_j(t) - \mathbf{s}_F(t)}{|\mathbf{s}_j(t) - \mathbf{s}_F(t)|^3} | dt \Rightarrow$$
   $$\Rightarrow dt(t) = \frac{\delta m_F^p}{\sum_{j \neq F} T((m_x m_y)^c, A_C(F,j)) | \frac{\mathbf{s}_j(t) - \mathbf{s}_F(t)}{|\mathbf{s}_j(t) - \mathbf{s}_F(t)|^3} |} \tag{6}$$

   It is apparent that for each t, $dt(t)$ is positive. The influence vector of each actor i is set:

   $$\mathbf{s}_i(t + dt(t)) = \mathbf{s}_i(t) + \frac{\mathbf{g}_i(t)}{m_i} \tag{7}$$

   (c) $t \leftarrow t + dt(t)$.

   (d) The test is executed inspecting whether there are actors i and j that meet the collision condition $(s_{i,j} \geq s_{j,j})$. If so, they fuse into one with new mass and influence vector as described above.

4.6. About the particle system contraction

Since a Markovian process is studied, where only the actual state of the system is taken into consideration, in order to prove the convergence of the algorithm it is enough to prove the contractiveness of the particles system. The authors in [37] proved the convergence of the original gravitational algorithm using an overlap function, G_s, instead of the product, using this formula for the attraction force between particles x and y is:

$$\mathbf{F}(x,y) = G_s(m_x m_y) \frac{\mathbf{s}_x - \mathbf{s}_y}{|\mathbf{s}_x - \mathbf{s}_y|^3}$$

In the case of the Borgia algorithm, the attraction force is the one described in Eq. (3). All the statements for the original attraction force formula still hold true for the new one. So, the convergence proof for the Borgia Algorithm is analogous to that already present in [37].

4.7. Parameter selection

The Borgia Clustering algorithm depends on a number of different parameters:

1. Affinity function: we can choose one or a combination of different affinity functions. Depending on the chosen one, we obtain different behaviours e.g. Best Friend affinity favours one-to-one interactions, while the Social Networking favours local high-density groups to attract.
2. Attraction force: we need to choose a T-norm and an exponent for the product of the masses. In the case of the T-norm, we can use the product as a default, and in case of the exponent, c, there are different alternatives: previous results in [28] indicates that $c = 0$ or $c = 1$ gives the best results in clustering.
3. Greedy expanse penalization factor: the bigger this factor, the harder it is for big masses to keep growing. Generally speaking, if we want to favour local interactions, we should set a high value ($+5$). If we want the big actors to take the lead in the process we should set a low value (0 or 1).

4.7.1. Affinity selection

We use the affinity function in the Borgia Clustering algorithm to reflect local pair-wise interactions and local group-level interactions. We reflect both characteristics using the best friend and the best common friend affinities (using the formulas in Table 1). By using the former, we can compute how affine two actors are. Based on the relationship between them; and using the latter, we can compute their affinity according to the common connections they share. To take both affinity functions into account, we have used a convex combination of them:

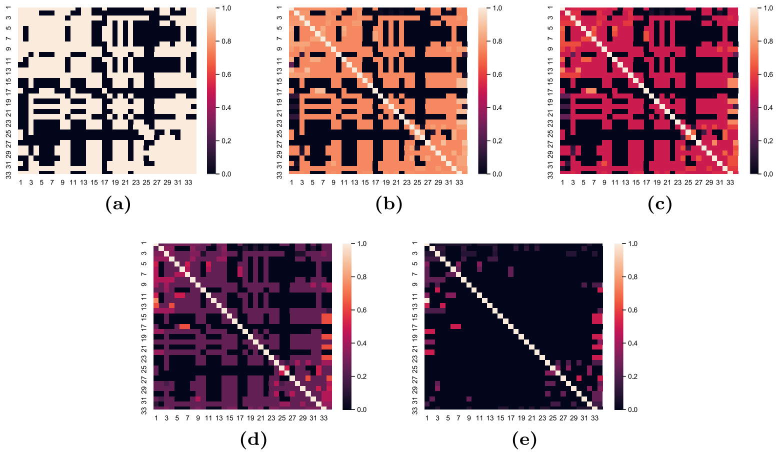
$$A_C(x,y) = \alpha BestFriend_C(x,y) + (1-\alpha) BestCommonFriend_C(x,y)$$

and then we choose and appropriate α. We have tested the effects of different α using the famous Zachary network [38], the co-occurrences character network of the Game of Thrones (GoT) novels [39] and the Eurovision voting phase [40]. A detailed description for these networks can be found in Sections 6 and 7.

In Fig. 5 we show the corresponding heatmap for each α and in Fig. 6 we show the corresponding network representations.

In Fig. 7 We check the effects of different α in the modularity value for the *GoT* and the Eurovision voting phase networks using a standard greedy expanse parameter and the exponent c.

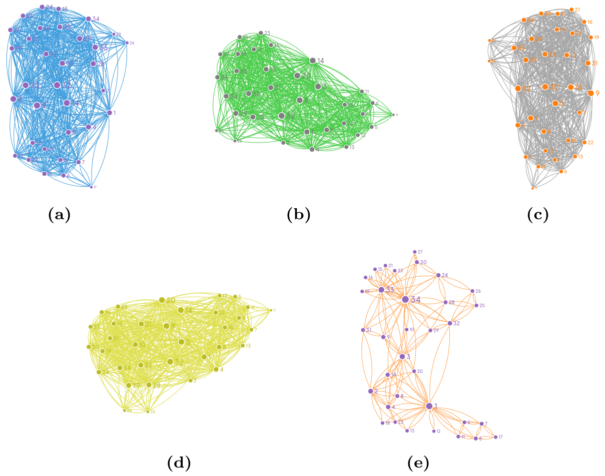
When it comes to the number of communities detected, there is no clear effect of the affinity used. It seems that using the Common Friend affinity in Eurovision, we get bigger communities, but this does not hold up in the *GoT* network.



(a) (b) (c)

(d) (e)

Fig. 5. Heatmap representation of the Zachary's karate social club. The S matrix of the Borgia algorithm with different α values. **a** $\alpha = 0.0$. **b** $\alpha = 0.25$ **c** $\alpha = 0.5$. **d** $\alpha = 0.75$. **e** $\alpha = 1.0$.



(a) (b) (c)

(d) (e)

Fig. 6. Network representation of the Zachary's karate social club Affinity Matrix. a $\alpha = 0.0$. **b** $\alpha = 0.25$ **c** $\alpha = 0.5$. **d** $\alpha = 0.75$. **e** $\alpha = 1.0$.

By measuring modularity and modularity density, it seems that a lower α benefits the modularity density, whilst a bigger value results in a better modularity. There is no α value that obtains the best results in both measures, but values in the interval [0.5, 1] apparently give a good result in terms of both metrics and the number of communities obtained.

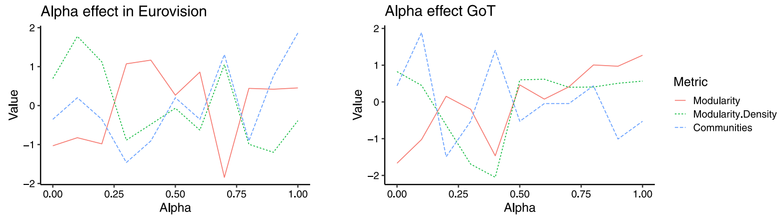
4.7.2. Greedy expanse parameter selection

We use the greedy expanse parameter, p, to limit the influence of big particles in the algorithm. The ideal value of the parameter might depend on the topology of the network and the rest of the chosen parameters. If p is too big, then the smaller communities will move too quickly, and if it is too small, the bigger communities will negate the local interactions of the smaller particles. In Fig. 8 we have studied how our algorithm performs using different p and different α values for the affinity function.

The chosen p value has a significant effect in the final result. Generally speaking, a correct value should be > 2, since it seems that the number of communities obtained in the ≤ 2 cases are abnormally high. The tendency in the modularity values is not so clear alongside p.

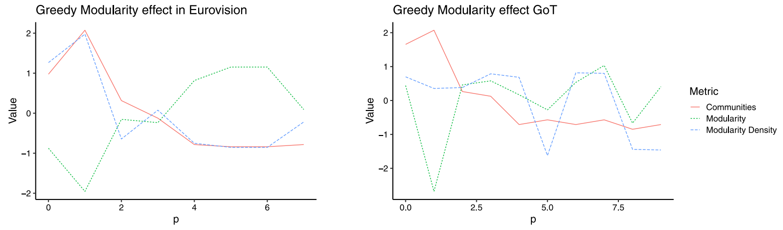
4.7.3. Delta parameter

The delta parameter delimits the maximal magnitude of movement for an actor in each iteration. In the classical gravitational algorithm, this parameter is a fixed number. However, our experimental tests have revealed that for two actors to collide for the first time we need much more iterations than for the rest of the collisions. We have tackled with this problem by setting a different delta for the first iteration, such that it warrants a collision in the first iteration. To do so, we calculate the distances



(a) (b)

Fig. 7. Affinity combination study a. Modularity values and number of communities for different combinations of Best Friend and Common Friend affinities in the Eurovision voting network with parameters $c = 0, p = 3$. **b.** Modularity values and number of communities for different combinations of Best Friend and Common Friend affinities in the *GoT* social network with parameters $c = 0, p = 3$. We have standardized the results for each metric.



(a) (b)

Fig. 8. Greedy parameter modularity study. Modularity [23] and Modularity Density [41] values and number of communities for different greedy expanse penalization parameter in the Eurovision voting network (**a**) and *GoT* (**b**). We have used the mean for three different α (0.5, 0.75, 1.0) in each p.

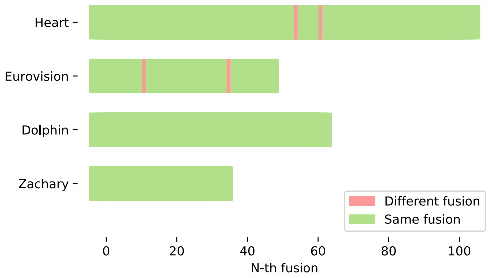


Fig. 9. Differences in the dynamic and static delta.

to collide for each pair of actors and then, set the delta value as the minimum of them.

The higher the delta, the less accurate the gravitational simulation is. The reason for this is that we only use this form of delta calculation in the first iteration. In this moment, actors are far from each other, which makes the attraction forces very weak.

In Fig. 9 we have studied the differences for four different datasets when using the static or the dynamic delta. There is no difference in two of them, while in the other two, only two fusions were different. The final result in all cases was not affected. However, changes in the execution time are consistently better (Fig. 10). It is important to note, however, that the simulation process in the Borgia Clustering can take longer execution time than other community detection algorithms.

5. The scalability problem

Social groups present lots of emergent properties when the number of people composing them increases. A group of fifty people is not just ten times a group of five, because human interaction does not follow a linear behaviour. In the case of the Borgia Clustering algorithm, we have used what we call "The Early Roman policy" [42] which aims at curbing the communities' tendency to grow as their size increases. In this context, we have implemented this policy by adding a "greedy expanse" penalization parameter, p, to correctly model the growth of social groups. This parameter penalizes "greedy" nodes that join too quickly with others due to their high attraction force. By setting a high p, they become slower, avoiding other nodes to join with one another so fast.

However, there is more than one way to cope with this additional complexity. Apart from the greedy expanse penalization, we have also proposed an additional a set of policies to treat the nodes labelled as too big (formed by the fusion of many actors) to be treated in the same way as smaller ones, which emerge as the result of different phenomena that still exist or have existed in human societies:

- Naive or linear policy: no difference between big and small communities.
- French or authoritarian policy: based on the famous French absolutist monarchies [43]. Only the affinities for the biggest actor in the node are taken into account.
- Early Roman policy: based on the end of the expansion of the Roman Empire [42]. Communities receive a penalization in their attraction power according to their size. This would be the case of the "greedy expanse" penalization parameter used in the experimental section.
- Late Roman policy: based on the Diocletian tetrarchy [44]. When a community is too big, it is divided in half depending on its members' affinities.

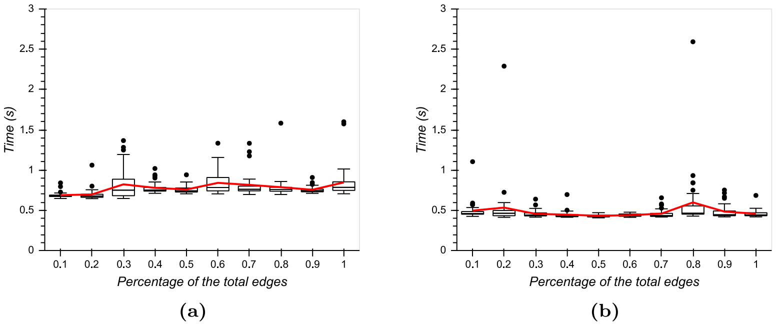


Fig. 10. Execution times for the Borgia Algorithm. a, b. Execution time for different number of edges in the full Eurovision network [40] (2369 total edges). The red line shows the mean of 30 executions of the algorithm, alongside the quartile distribution, for different network sizes. These networks have been obtained by randomly sampling the desired percentage of edges from the original graph. While (a) shows the times for the static delta, (b) shows the times for the dynamic delta. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- Greek policy: based on the organization of the ancient Greek world [45]. When two actors are about to form a big node, they are frozen instead and they cannot have any more attractive interactions between them.
- Aristocratic policy: based on the natural hierarchy formed in many societies [46]. When a community is too big, it is treated as a smaller community formed only by the top-k most important members of the community.

6. Results for real-life networks

In this section we present the results for the community detection in three different social networks: *GoT*, Eurovision and the word association network for *Heart of Darkness*. The cut for each dendrogram has been obtained by using the best configuration according to the formula in Eq. (4).

6.1. Game of thrones

Song of Ice and Fire is a popular fantasy book in the saga written by George R.R. Martin which, to this day, is constituted by five books. This series of novels presents a numerous set of interesting characters, most of them members of one of the dynasties that rule the fantasy world of Westeros.

We used the data in [47], that counts co-occurrences of each character in the original text, and then we created the adjacency matrix.

To compute the algorithm on *Song of Ice Fire* network we used $\alpha = 0.5$, $p = 3$ and $c = 0$.

The result in Fig. 11 shows how the characters are grouped around the different sub-plots in the book. If we look at the dendrogram, we can see that the characters whose plot is more “stable” (they do not change much from places or acquaintances) form or join communities faster than those who play an important role in different places or events in the books.

6.2. Eurovision song contest

Eurovision is a musical contest where 42 countries compete against each other in order to get the highest score. The participants should judge and give points to the rest according to the quality of the musical performance. This is, of course, only the theory, since in practice it is noticeable how cultural and geographical proximities play an important role in the outcome of the contest. From the dataset in [40], we have taken not only the last decade of voting, but also both the whole record of finals voting to study how countries historically behave when they are voting. These networks are interesting to us because they have very high density (Fig. 12), as they are almost fully connected, and because we actually have semantic information about each actor. This means that we can interpret the resulting communities without relying too much on numerical measures.

In this case, we use the same parameter selection as in the previous case; $\alpha = 1$, $p = 3$ and $c = 0$.

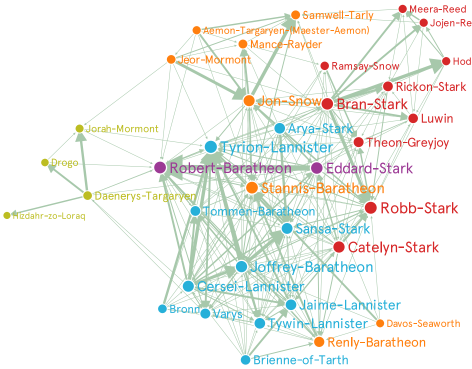
In the first case, the last decade of Eurovision (Figs. 12a, 13a), we have obtained four different communities. At a local level, it seems that strong cultural and geographical components influence the voting process, e.g Greece and Cyprus, Spain and Portugal. These components are more relevant in some communities than in others. There is a community consisting almost exclusively of countries of Slavic Europe, other has mainly Mediterranean countries, etc. However, in our time cultural bias is not so pronounced when compared to the whole historic record of voting.

If we take into account the whole voting record since 1975, we obtain five communities (Figs. 12b, 13b). Two of them have a clear cultural resemblance between their members: the Nordic countries, and the Spain–France–Portugal–Andorra axis. There are also two Eastern-Europe groups and a Western one (with some exceptions in them that could be a consequence of a “migration” effect [48]).

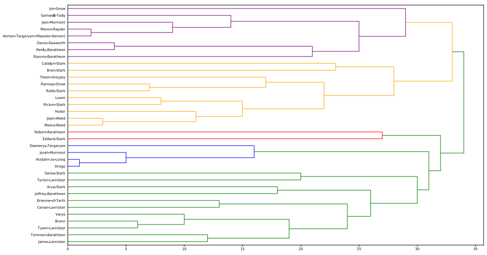
6.3. Heart of darkness

Heart of Darkness is the famous novel written by Joseph Conrad in 1899. The story narrates the abuse committed by the Belgium King during the late 19th century. Besides its humanistic interest, this novel presents a mysterious character, Kurtz, which is particularly interesting for our research. This omnipresent individual, seen as a demigod by the natives who constantly refer to him, only makes an appearance in the final stages of the book. We have computed the word association network in the same way as that in Section 3. Each node corresponds to a word in the original text and each edge is the corresponding affinity between two entities, using as C the original number of co-occurrences for each pair of words in the same paragraph.

We can see the result using Borgia Clustering community detection in Fig. 14. We obtain three different communities. The most notable one is the “Kurtz” community. It is characterized by the number of words that induce a moral bias in the reader about



(a)



(b)

Fig. 11. Community detection in the *GoT* network. a. The network formed by the top-35 most important characters in the book. Each of the five communities is marked with a different colour. **b**. The dendrogram formed during the execution of the algorithm. Each community is marked with same colours as in (a). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

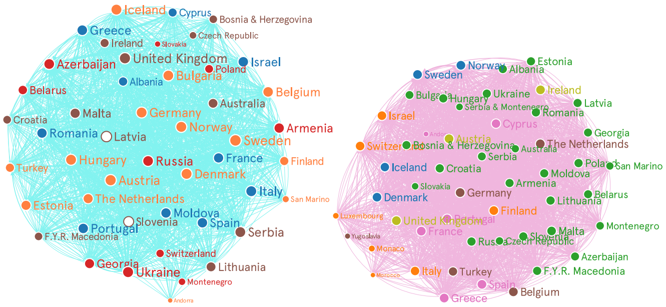


Fig. 12. Networks and community detection in Eurovision contest. a. Last decade of voting record in Eurovision finals. **b** Whole voting record in Eurovision finals.

this character, such as: ''Darkness'', ''Devil'', or ''Power''. There are also some other important concepts linked to this character: ''Knowledge'', ''Desire'', ''Reason'', ''Eloquence'', etc. All of these show the fascination that his figure inspires in the natives that the first-person narrator and protagonist encounters through the journey. The ''man'' community and the ''time'' are both semantically more heterogeneous, although the latter one is mainly composed of terms that describe the place in which the action takes place.

7. Comparison with other community detection algorithms

Finally, to benchmark our community detection algorithm, we have chosen datasets with ground truth labels: the famous Zachary's karate club social network [38], politics books [49], that contains the number of co-purchases of different books about US politics, football network [17], that represents the number of matches between each pair of teams, and Dolphins [50] which is a network that registers the frequency each pair of dolphins played together.

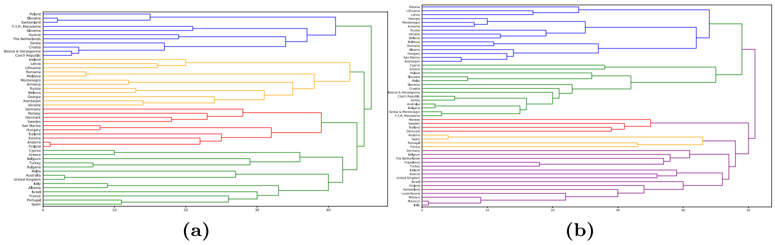
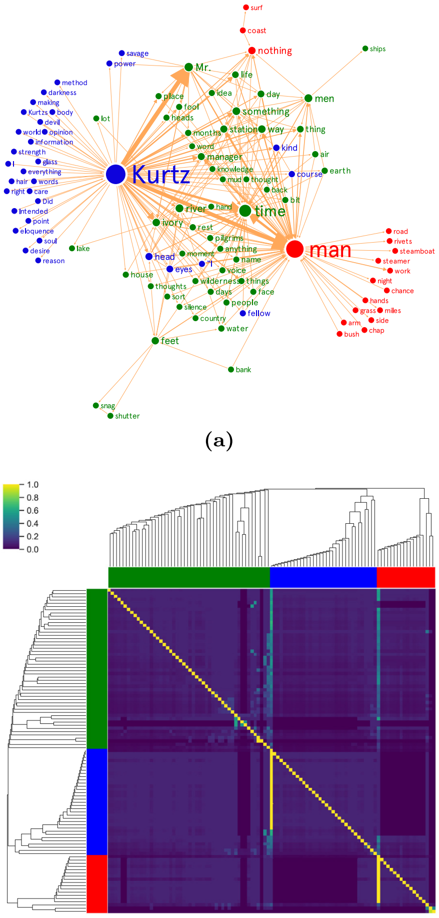

(a) (b)

Fig. 13. Dendrogram and community detection for the last decade of voting record in Eurovision finals. a. Dendrogram and community detection for the last decade of voting record in Eurovision finals. **b** Dendrogram community detection for the whole voting record in Eurovision finals.



(a)

(b)

Fig. 14. Community detection in *Heart of Darkness*. a. The network resulting from the top-200 most repeated names in the book. Each one of the five communities is marked with a different colour. **b**. A_C heatmap and the dendrogram formed during the execution of the algorithm. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Results for the Borgia Clustering. a. Zachary Karate club network. Parameters: $\alpha = 0.7, p = 3, G(x, y) = 1$. **b.** Dolphin network. Parameters: $\alpha = 0.7, p = 3, G(x, y) = 1$. **c.** Football network. Parameters: $\alpha = 1, p = 0, G(x, y) = 1$. **d.** Polbooks club network. Parameters: $\alpha = 1.0, p = 0, G(x, y) = 1$.

	Mod.	M. density	ARI	NMI	C	Mod.	M. density	ARI	NMI	C
Greedy Mod.	0.3871	0.1868	0.5684	0.4540	3	0.4954	0.1659	0.4658	0.4149	4
Girvan–Newman	**0.4156**	0.2258	0.4070	0.3529	4	**0.5193**	0.2011	0.4505	0.4404	5
Label prop.	0.3956	0.2139	0.6841	0.5750	3	0.4684	0.1829	0.492	0.4418	4
L. eigenvector	0.4012	0.1969	0.4351	0.4172	4	0.4911	0.1809	0.3211	0.3439	5
Lovaine	0.4138	**0.2296**	0.4292	0.3592	4	0.5175	**0.2032**	0.3140	0.3397	5
Borgia C.	0.3693	0.1797	**0.8822**	**0.8324**	2	0.3787	0.1362	**1.0000**	**1.0000**	2
Grav. Clus.	0.0342	0.0555	0.0325	0.0757	3	−0.0010	0.0422	0.0091	0.0115	4
		(a)						(b)		

	Mod.	M. density	ARI	NMI	C	Mod.	M. density	ARI	NMI	C
Greedy Mod.	0.5564	0.2744	0.4844	0.5584	6	0.5019	0.1739	0.6378	0.4929	4
Girvan–Newman	0.5996	0.4321	0.7781	0.8014	10	0.5168	0.1989	**0.6823**	0.4875	5
Label prop.	0.5974	0.4027	0.7445	0.7701	9	0.5106	0.1874	0.6701	0.5088	4
L. eigenvector	0.4926	0.2593	0.4640	0.5611	8	0.4671	0.1439	0.5466	0.4428	4
Lovaine	**0.6044**	0.4167	0.7070	0.7551	9	**0.5267**	**0.1991**	0.6463	0.4576	5
Borgia C.	0.6005	**0.4909**	**0.8966**	**0.8978**	12	0.4994	0.1665	0.6685	**0.5649**	3
Grav. Clus.	0.2319	0.0848	0.1202	0.2899	7	0.0053	0.0646	0.0234	0.0203	2
		(c)						(d)		

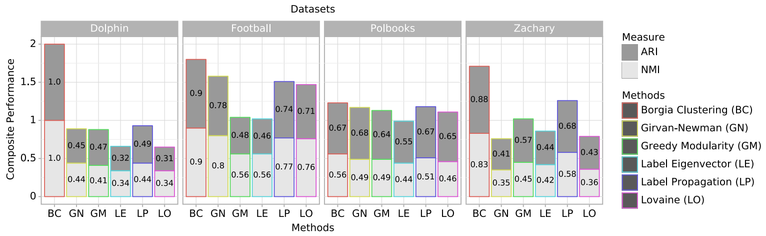


Fig. 15. Comparison of different Community Detection Algorithms. We have compared the Borgia Clustering algorithm against three modularity optimization methods: Girvan–Newman, Newman greedy modularity optimization, and the Lovaine algorithm. We have also compared it against using eigenvalues of matrices to detect communities and label propagation.

To test the quality of our solution and to study the best parameter selection in the Borgia Clustering algorithm, we have compared it to other community detection algorithms: Girvan–Newman [20], Newman greedy modularity optimization [24], the Lovaine algorithm [25], using the eigenvalues of matrices to detect communities [51], and label propagation [52]. The reported results for the label propagation method is the median of five executions due to its stochastic nature. We have used the Normalized Mutual Information (NMI) [53] and Random Adjusted Index (ARI) [54] to compare the results against ground truth labels. We have also compared Modularity [23] and Modularity Density [41] for each solution.

The results are shown in Fig. 15 using the Borgia Clustering algorithm in the ARI and NMI measures. A comparison including modularity and modularity density can be found in Table 3. Borgia Clustering does not use at all the concept of modularity, so it was expected to perform worse with respect to this index compared to the methods that actually optimize this metric: Greedy Modularity, Girvan–Newman and Lovaine.

In the same table, we also show the results for the traditional gravitational algorithm that was developed originally for clustering problems [28]. As expected, this algorithm performs poorly compared to the others, as it was not specifically designed for this problem. Using the affinity matrices instead of the adjacency matrices showed no improvement for this algorithm either.

8. Conclusions and future work

In this work we have proposed a new set of functions based on human nature to represent actors and we have shown the effects that different affinities have on the same real-work networks. We have discussed the importance of scale in networks, and we have applied our ideas to develop a new human-based community detection algorithm, the Borgia Clustering, based on the centre-Italian wars of the 15th century. Using this algorithm, we have obtained good results compared to the most used community detection algorithms up-to-date. We think that this algorithm can offer significant improvements in very dense networks, when modularity is not a reliable measure to optimize. Also, this algorithm always converges to the same value, so there is no need to run it multiple times to obtain a valid result. Furthermore, due to the intuitive principles that the algorithm stands on, we think that results obtained with it can be easily interpretable.

Future research will aim at applying Borgia Clustering in large scale networks, studying the constraints and requirements to perform the community detection process efficiently, in terms of memory and execution time. We also intend to explore the effects of affinity functions in dynamic environments, where networks change along time [55], and exploit them with the Borgia Clustering algorithm.

CRediT authorship contribution statement

J. Fumanal-Idocin: Conceptualization, Software, Writing - original draft, Investigation. **A. Alonso-Betanzos:** Methodology, Writing - review & editing, Resources. **O. Cordón:** Methodology, Writing - review & editing, Resources. **H. Bustince:** Writing - review & editing, Project administration, Funding acquisition. **M. Minárová:** Writing - review & editing, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Javier Fumanal Idocin's and Humberto Bustince's research has been supported by the project TIN2016-77356-P (AEI/FEDER,UE).

Oscar Cordón's research was supported by the Spanish Ministry of Science, Innovation and Universities under grant EXASOCO (PGC2018-101216-B-I00), including, European Regional Development Funds (ERDF).

Amparo Alonso-Betanzos' research has been financially supported in part by the Spanish Ministerio de Economía y Competitividad (research project TIN2015-65069-C2-1-R), by European Union FEDER funds and by the Consellería de Industria of the Xunta de Galicia, Spain (research project GRC2014 /035).

M. Minárová's research has been funded by the project work was supported by the project APVV-17-0066.

References

[1] J. Scott, Social network analysis, Sociology 22 (1) (1988) 109–127.
[2] S. Wasserman, K. Faust, et al., Social Network Analysis: Methods and Applications, Vol. 8, Cambridge University Press, 1994.
[3] S.P. Borgatti, A. Mehra, D.J. Brass, G. Labianca, Network analysis in the social sciences, Science 323 (5916) (2009) 892–895.
[4] S. Horvath, Weighted Network Analysis: Applications in Genomics and Systems Biology, Springer Science & Business Media, 2011.
[5] J.A. Benítez-Andrades, I. García-Rodríguez, C. Benavides, H. Alaiz-Moretón, A. Rodríguez-González, Social network analysis for personalized characterization and risk assessment of alcohol use disorders in adolescents using semantic technologies, Future Gener. Comput. Syst. 106 (2020) 154–170.
[6] C.S. Fischer, To Dwell Among Friends: Personal Networks in Town and City, University of Chicago Press, 1982.
[7] B. Wellman, S.D. Berkowitz, Social Structures: A Network Approach, Vol. 2, CUP Archive, 1988.
[8] A. Stopczynski, V. Sekara, P. Sapiezynski, A. Cuttone, M.M. Madsen, J.E. Larsen, S. Lehmann, Measuring large-scale social networks with high resolution, PLoS One 9 (4) (2014).
[9] N. Sánchez-Maroño, A. Alonso-Betanzos, O. Fontenla-Romero, C. Brinquis-Núñez, J.G. Polhill, T. Craig, A. Dumitru, R. García-Mira, An agent-based model for simulating environmental behavior in an educational organization, Neural Process. Lett. 42 (1) (2015) 89–118.
[10] C. Gunaratne, C. Senevirathna, C. Jayalath, N. Baral, W. Rand, I. Garibay, A Multi-action cascade model of conversation, in: 5th International Conference on Computational Social Science, 2019.
[11] I. Lobel, E. Sadler, Information diffusion in networks through social learning, Theor. Econ. 10 (3) (2015) 807–851.
[12] J. McAuley, J. Leskovec, Discovering social circles in ego networks, ACM Trans. Knowl. Discov. Data 8 (1) (2014) 4.
[13] L. Sless, N. Hazon, S. Kraus, M. Wooldridge, Forming k coalitions and facilitating relationships in social networks, Artificial Intelligence 259 (2018) 217–245.
[14] W. Yu, S. Li, Recommender systems based on multiple social networks correlation, Future Gener. Comput. Syst. 87 (2018) 312–327.
[15] X. Luo, C. Jiang, W. Wang, Y. Xu, J.-H. Wang, W. Zhao, User behavior prediction in social networks using weighted extreme learning machine with distribution optimization, Future Gener. Comput. Syst. 93 (2019) 1023–1035.
[16] J. Delgado, Emergence of social conventions in complex networks, Artificial Intelligence 141 (1) (2002) 171–185.
[17] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
[18] H. Zhou, Distance, dissimilarity index, and network community structure, Phys. Rev. E 67 (6) (2003) 061901.
[19] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814.
[20] M. Girvan, M.E. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
[21] C. Pizzuti, Ga-net: A genetic algorithm for community detection in social networks, in: International Conference on Parallel Problem Solving from Nature, Springer, 2008, pp. 1081–1090.
[22] T. Ma, Q. Liu, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, LGIEM: Global and local node influence based community detection, Future Gener. Comput. Syst. 105 (2020) 533–546.
[23] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.
[24] M.E. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.
[25] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008).
[26] L. Danon, A. Diaz-Guilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. Theory Exp. 2005 (09) (2005) P09008.
[27] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.
[28] W. Wright, Gravitational clustering, Pattern Recognit. 9 (3) (1977) 151–166.
[29] G. Beliakov, H.B. Sola, T.C. Sánchez, A Practical Guide to Averaging Functions, Vol. 329, Springer, 2016.
[30] S. Milgram, The small world problem, Psychol. Today 2 (1) (1967) 60–67.
[31] N.E. Friedkin, Theoretical foundations for centrality measures, Am. J. Sociol. 96 (6) (1991) 1478–1504.
[32] M. Jacomy, T. Venturini, S. Heymann, M. Bastian, ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software, PLoS One 9 (6) (2014) 1–12.
[33] Kayac, Italian lordships in 1499, 2013, URL https://commons.wikimedia.org/wiki/File:Grandi_Casate_Italiane_nel_1499.png. (Online; Accessed 20 March 2019).
[34] J. Catalàn Deus, El príncipe del renacimiento, Debate 26 (2008) 90.
[35] R.K. Merton, The Matthew effect in science: The reward and communication systems of science are considered, Science 159 (3810) (1968) 56–63.
[36] M. Gupta, J. Qi, Theory of T-norms and fuzzy inference methods, Fuzzy Sets and Systems 40 (3) (1991) 431–450.
[37] J. Armentia, I. Rodrí-guez, J. Fumanal Idocin, H. Bustince, M. Minárová, A. Jurio, Gravitational clustering algorithm generalization by using an aggregation of masses in Newton law, in: R. Halaš, M. Gagolewski, R. Mesiar (Eds.), New Trends in Aggregation Theory, Springer International Publishing, Cham, 2019, pp. 172–182.
[38] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.
[39] mathbeveridge, Asoiaf, GitHub, 2017, URL https://github.com/mathbeveridge/asoiaf. (Online; Accessed 26 September 2019).
[40] Eurovision Song Contest scores 1975–2018, 2019, URL https://www.kaggle.com/datagraver/eurovision-song-contest-scores-19752018. (Online; Accessed 2 July 2019).
[41] Z. Li, S. Zhang, R.-S. Wang, X.-S. Zhang, L. Chen, Quantitative function for community detection, Phys. Rev. E 77 (3) (2008) 036109.
[42] M. Fulford, Territorial expansion and the Roman Empire, World Archaeol. 23 (3) (1992) 294–305.
[43] F. Cosandey, R. Descimon, L'absolutisme en France: histoire et historiographie, Editions du Seuil, 2002 (In French).
[44] R. Rees, Diocletian and the Tetrarchy, Edinburgh University Press Edinburgh, 2004.
[45] M.H. Hansen, Polis: An Introduction to the Ancient Greek City-State, Oxford University Press, 2006.
[46] J. Gledhill, B. Bender, M.T. Larsen, State And Society: The Emergence and Development of Social Hierarchy and Political Centralization, Vol. 4, Psychology Press, 1995.
[47] A. Beveridge, M. Chemers, The game of game of thrones: Networked concordances and fractal dramaturgy, in: Reading Contemporary Serial Television Universes, Routledge, 2018, pp. 201–225.
[48] L. Spierdijk, M. Vellekoop, Geography, Culture, and Religion: Explaining the Bias in Eurovision Song Contest Voting, Department of Applied Mathematics, University of Twente, Enschede, 2006, p. 33.
[49] L.A. Adamic, N. Glance, The political blogosphere and the 2004 US election: divided they blog, in: Proceedings of the 3rd International Workshop on Link Discovery, ACM, 2005, pp. 36–43.
[50] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. 54 (4) (2003) 396–405.
[51] M.E. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006) 036104.
[52] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.
[53] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (Dec) (2002) 583–617.
[54] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Amer. Statist. Assoc. 66 (336) (1971) 846–850.
[55] D. Greene, D. Doyle, P. Cunningham, Tracking the evolution of communities in dynamic social networks, in: 2010 International Conference on Advances in Social Networks Analysis and Mining, 2010, pp. 176–183.

Javier Fumanal Idocin holds a B.Sc in Computer Science at the University of Zaragoza, Spain and a M.Sc in Data Science and Computer Engineering at the University of Granada, Spain. He is now a Ph.D. Student of the Public University of Navarre, Spain in the department of Statistics, Informatics and Mathematics. His research interests include machine intelligence, fuzzy logic, social networks and Brain–Computer Interfaces.

Amparo Alonso-Betanzos is a Full Professor at the Department of Computer Science and coordinator of the Laboratory for R&D in Artificial Intelligence at the ICT Center (CITIC) of the University of A Coruña (UDC). She is also the President of the Spanish Association for Artificial Intelligence (AEPIA). Currently, she is working on the development of Scalable Machine Learning algorithms, and on their applications to several fields. Another fields of interest are trustworthy AI, mainly in privacy by design algorithms and Explainable AI.

Oscar Cordón is Full Professor with the University of Granada (UGR), Spain. He was the founder and leader of its Virtual Learning Center (2001–05) and the Vice-President for Digital University (2015–19). He also was one of the founding researchers of the European Centre for Soft Computing (2006–2011), being later contracted as Distinguished Affiliated Researcher (2011–15).

He has been, for more than 25 years, an internationally recognized contributor to R&D Programs in fundamentals and real-world applications of computational intelligence. He has published >360 peer-reviewed scientific publications including a research book on genetic fuzzy systems (with >1300 citations in Google Scholar) and 105 JCR-SCI-indexed journal papers (62 in Q1), advised 19 Ph.D. dissertations, and coordinated 36 research projects and contracts (with an overall amount of >9M€). By December 2019, his publications had received 4866 citations (h-index=35), being included in the 1 and h-index=54 in Google Scholar). He also has a granted international patent on an intelligent system for forensic identification commercialized in Mexico and South Africa.

Prof. Cordón received the UGR Young Researcher Career Award in 2004; the IEEE Computational Intelligence Society (CIS) Outstanding Early Career Award in 2011 (the first such award conferred); the IFSA Award for Outstanding Applications of Fuzzy Technology in 2011; and the National Award on Computer Science ARITMEL by the Spanish Computer Science Scientific Society in 2014. He was elevated to IEEE Fellow in 2018 and received the IFSA Fellow in 2019. He is currently or was Associate Editor of 18 international journals, and was recognized as IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding AE in 2008. Since 2004, he has taken many different representative positions with EUSFLAT and the IEEE CIS.

Humberto Bustince received the Graduate degree in physics from the University of Salamanca in 1983 and Ph.D. in mathematics from the Public University of Navarra, Pamplona, Spain, in 1994. He is a Full Professor of Computer Science and Artificial Intelligence in the Public University of Navarra, Pamplona, Spain where he is the main researcher of the Artificial Intelligence and Approximate Reasoning group, whose main research lines are both theoretical (aggregation functions, information and comparison measures, fuzzy sets, and extensions) and applied (image processing, classification, machine learning, data mining, and big data). He has led 11 I+D public-funded research projects, at a national and at a regional level. He is currently the main researcher of a project in the Spanish Science Program and of a scientific network about fuzzy logic and soft computing. He has been in charge of research projects collaborating with private companies. He has taken part in two international research projects. He has authored more than 210 works, according to Web of Science, in conferences and international journals, with around 110 of them in journals of the first quartile of JCR. Moreover, five of these works are also among the highly cited papers of the last ten years, according to Science Essential Indicators of Web of Science. Dr. Bustince is the Editor-in-Chief of the online magazine Mathware & Soft Computing of the European Society for Fuzzy Logic and technologies and of the Axioms journal. He is an Associated Editor of the IEEE Transactions on Fuzzy Systems Journal and a member of the editorial board of the Journals Fuzzy Sets and Systems, Information Fusion, International Journal of Computational Intelligence Systems and Journal of Intelligent & Fuzzy Systems. He is the coauthor of a monography about averaging functions and coeditor of several books. He has organized some renowned international conferences such as EUROFUSE 2009 and AGOP. Honorary Professor at the University of Nottingham, National Spanish Computer Science Award in 2019 and EUSFLAT Excellence Research Award in 2019.

Mária Minárová graduated from the Faculty of Mathematics and Physics, Comenius University Bratislava, receiving her master degree in mathematics. Since 1991 she works at the Department of Mathematics and Descriptive Geometry at the Faculty of Civil Engineering at the Slovak University of Technology, where she received her Ph.D. degree in 2004. She deals with applied mathematics, her research is focused at mathematical modelling in building physics, rheology, biomechanics, biomedicine and data mining.

6.8 A Generalization of the Sugeno integral to aggregate Interval-valued data: an application to Brain Computer Interface and Social Network Analysis

Associated publication:

- Fumanal-Idocin J., Takac Z., Horanska L., da Cruz Asmus T., Dimuro G., Vidaurre C., Fernandez J. & Bustince H. (2022). A Generalization of the Sugeno integral to aggregate Interval-valued data: an application to Brain Computer Interface and Social Network Analysis. *Fuzzy Sets & Systems.*

  Status: Published.

  Impact Factor (JCR 2021): 4.462

  Categories:

  Computer Science, Theory & methods. Ranking 19/109 (Q1).

  Mathematics, applied. Ranking 6/267 (Q1).

  Statistics & Probability. Ranking 9/125 (Q1).

Available online at www.sciencedirect.com

ScienceDirect

Fuzzy Sets and Systems 451 (2022) 320–341

FUZZY sets and systems

www.elsevier.com/locate/fss

A generalization of the Sugeno integral to aggregate interval-valued data: An application to brain computer interface and social network analysis

J. Fumanal-Idocin [a,*], Z. Takáč [b], L'. Horanská [b], T. da Cruz Asmus [d], G. Dimuro [c], C. Vidaurre [e], J. Fernandez [a], H. Bustince [a,*]

[a] *Public University of Navarra and Institute of Smart Cities, Campus Arrosadia s/n, 31006 Pamplona, Spain*
[b] *Institute of Information Engineering, Automation and Mathematics, Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Radlinského 9, Bratislava, 812 37, Slovak Republic*
[c] *Centro de Ciencias Computacionais, Universidade Federal do Rio Grande, Av. Itália km 08, Campus Carreiros, Rio Grande, Brazil*
[d] *Instituto de Matemática, Estatística e Física, Universidade Federal do Rio Grande, Av Italia km 08, Campus Carreiros, Rio Grande, Brazil*
[e] *Parque Científico y Tecnológico de Guipúzcoa, Paseo Mikeletegi 2, 20009 San Sebastian, Spain*

Received 9 December 2021; received in revised form 3 October 2022; accepted 6 October 2022
Available online 10 October 2022

Abstract

Intervals are a popular way to represent the uncertainty related to data, in which we express the vagueness of each observation as the width of the interval. However, when using intervals for this purpose, we need to use the appropriate set of mathematical tools to work with. This can be problematic due to the scarcity and complexity of interval-valued functions in comparison with the numerical ones. In this work, we propose to extend a generalization of the Sugeno integral to work with interval-valued data. Then, we use this integral to aggregate interval-valued data in two different settings: first, we study the use of intervals in a brain-computer interface; secondly, we study how to construct interval-valued relationships in a social network, and how to aggregate their information. Our results show that interval-valued data can effectively model some of the uncertainty and coalitions of the data in both cases. For the case of brain-computer interface, we found that our results surpassed the results of other interval-valued functions.

Keywords: Aggregation function; Sugeno integral; Generalized Sugeno integral; Brain computer interface; Social network

[*] Corresponding authors.
E-mail addresses: javier.fumanal@unavarra.es (J. Fumanal-Idocin), zdenko.takac@stuba.sk (Z. Takáč), lubomira.horanska@stuba.sk (L'. Horanská), gracaliz.pereira@unavarra.es (G. Dimuro), carmen.vidaurre@tecnalia.com (C. Vidaurre), fcojavier.fernandez@unavarra.es (J. Fernandez), bustince@unavarra.es (H. Bustince).

https://doi.org/10.1016/j.fss.2022.10.003
0165-0114/© 2022 Published by Elsevier B.V.

1. Introduction

Aggregation functions are used to fuse several data into one single output value, representative of the original set of inputs [1,2]. As this fusion process is critical in many multi-component systems, aggregation functions have been studied in a wide range of different settings, like fuzzy rule-based classification systems [3–5], image processing [6,7], unsupervised learning [8], decision making [9,10,6] or brain computer-interfaces [11,12].

Two of the most popular aggregation functions are the Choquet and Sugeno integrals [13,27]. These integrals fuse the data using a fuzzy measure, which models the different relations among the inputs [14,15]. By using these functions we are able to model the coalitions between the different features to fuse in the aggregation process. The famous Ordered Weighted Aggregation operators (OWAs) [16] are a particular case of the Choquet integral using a symmetric fuzzy measure [17].

Both the Choquet and Sugeno integrals have been profoundly studied, and many works have been published regarding their characterization, properties, and relation with other aggregation functions [18–21]. More recently, a series of generalizations of the Choquet integral have been proposed and applied to different fuzzy-rule based classification systems [19,3,15]. The Sugeno integral has been generalized in a similar way, applied to image processing [25].

However, all of these works have been proposed to deal with numerical data. The use of other data representations, like intervals, have been proven popular to model the uncertainty linked to experimental observations or estimations, but it requires some additional challenges related to the lack of standard order in the framework of intervals [22]. An Interval-Valued Sugeno integral (IV-Sugeno) was proposed in [23], and some more general expressions encompassing both Choquet and Sugeno integral were given in [24]. However, the different generalizations of the Sugeno integral were not studied. We believe that these generalizations can be of interest in the interval-valued setting, as its numerical counterpart has been successfully applied to Brain Computer Interface systems [12] and image thresholding [25].

In this work, we present an in-depth study of the properties of the IV-Sugeno, and its possible generalizations. Then, we present two different applications using interval-valued data:

- We present a novel way to construct intervals in a Motor Imagery (MI) Brain Computer Interface (BCI) classification framework. We explain how the intervals are constructed from the output of different classifiers, and how we aggregate them using the IV-Sugeno integral.
- We present the interval-valued version of affinity functions in social network analysis [26]. These functions use intervals to measure the difference in commitment in a pairwise relationship between two people. Then, we explain how we use the IV-Sugeno to characterize each actor in the network based how asymmetric are its relationships.

The rest of the paper goes as follows. Section 2 recalls some notions needed to understand the rest of the paper: aggregation theory, the numerical Sugeno integral and interval-valued aggregations. In Section 3, we introduce so-called interval-valued Sugeno-like FG-functional as an interval-valued generalization of the Sugeno integral. In Section 4, the mathematical properties of the proposed functionals are studied, and in Section 5, a construction method for interval-valued Sugeno-like FG-functionals is proposed. Section 6 illustrates the proposed BCI framework and its use of interval-valued Sugeno-like FG-functionals. Section 7 proposes the interval-valued affinity functions, and four centrality measures for social network analysis using the newly proposed ideas. Finally, in Section 8, we give our conclusions and future lines for this work.

2. Preliminary

In this section we recall some notions needed in our subsequent developments. We also fix the notation, mostly in accordance with [2], wherein more details on theory of aggregation functions can be found.

2.1. Aggregation functions

The process of merging an information represented by several values into a single one is formalized by so-called aggregation functions. The finite space of attributes can be represented by the set $N = \{1, \dots, n\}$, $n \in \mathbb{N}$ and the inputs by n-tuples of reals from the unit interval $[0, 1]$. Let us denote vectors $(x_1, \dots, x_n) \in [0, 1]^n$ by bold symbols $\mathbf{x}$.

Definition 1. Let $n \in \mathbb{N}$. A function $A\colon [0,1]^n \to [0,1]$ is an n-ary aggregation function if it is nondecreasing in each variable and the boundary conditions $A(0,\dots,0) = 0$ and $A(1,\dots,1) = 1$ are satisfied.

We list some of the well-known aggregation functions. Let $\mathbf{x} \in [0,1]^n$.

- The *arithmetic mean* $\mathrm{AM}_{(n)}$ is defined by $\mathrm{AM}_{(n)}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i$.
- The *weighted arithmetic mean* $\mathrm{WAM}_{\mathbf{w}}$ is defined by $\mathrm{WAM}_{\mathbf{w}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} w_i x_i$, where the weight vector $\mathbf{w} = (w_1,\dots,w_n) \in [0,1]^n$ is such that $\sum_{i=1}^{n} w_i = 1$.
- The *minimum* and *maximum* operators defined by $\mathrm{Min}(\mathbf{x}) = \min\{x_1,\dots,x_n\}$ and $\mathrm{Max}(\mathbf{x}) = \max\{x_1,\dots,x_n\}$, respectively, can be regarded as the special cases of the so-called *order statistics* $\mathrm{OS}_k\colon [0,1]^n \to [0,1]$ defined by $\mathrm{OS}_k(\mathbf{x}) = x_{\sigma(k)}$, where σ is a permutation on N such that $x_{\sigma(1)} \le \dots \le x_{\sigma(n)}$. Clearly, $\mathrm{Min} = \mathrm{OS}_1$ and $\mathrm{Max} = \mathrm{OS}_n$.

The following properties of real functions can be desirable in different contexts of aggregation.

Definition 2. Let $n \in \mathbb{N}$. A function $A\colon [0,1]^n \to \mathbb{R}$ is said to be:

- *internal*, if for each $\mathbf{x} \in [0,1]^n$ it holds $\mathrm{Min}(\mathbf{x}) \le A(\mathbf{x}) \le \mathrm{Max}(\mathbf{x})$;
- *idempotent*, if $A(x,\dots,x) = x$ for each $x \in [0,1]$;
- *comonotone maxitive*, if $A(\mathbf{x} \vee \overline{\mathbf{x}}) = A(\mathbf{x}) \vee A(\overline{\mathbf{x}})$ for all comonotone vectors $\mathbf{x}, \overline{\mathbf{x}} \in [0,1]^n$ (vectors $\mathbf{x}, \overline{\mathbf{x}}$ are comonotone, if $(x_i - x_j)(\overline{x}_i - \overline{x}_j) \ge 0$ for all $i, j \in \{1,\dots,n\}$);
- *comonotone minitive*, if $A(\mathbf{x} \wedge \overline{\mathbf{x}}) = A(\mathbf{x}) \wedge A(\overline{\mathbf{x}})$ for all comonotone vectors $\mathbf{x}, \overline{\mathbf{x}} \in [0,1]^n$;
- *positively homogeneous*, if $A(c\mathbf{x}) = cA(\mathbf{x})$ for each $\mathbf{x} \in [0,1]^n$ and $c > 0$ such that $c\mathbf{x} \in [0,1]^n$;
- *min-homogeneous*, if $A(c \wedge \mathbf{x}) = c \wedge A(\mathbf{x})$ for each $\mathbf{x} \in [0,1]^n$ and $c \in [0,1]$.

2.2. Sugeno integral

The Sugeno integral introduced in 1974 [27] is widely used in many applications due to its ability to model interactions between inputs by means of the so-called fuzzy measure.

Definition 3. A set function $m\colon 2^N \to [0,1]$ is a fuzzy measure, if $m(C) \le m(D)$ whenever $C \subseteq D \subseteq N$ and $m(\emptyset) = 0$, $m(N) = 1$.

Definition 4. A fuzzy measure m is said to be *symmetric, if $m(E) = m(F)$ whenever $|E| = |F|$ for all $E, F \subseteq N$ (here $|E|$ stands for the cardinality of the set E).*

Restricting to the finite space and vectors of reals from the unit interval, the (discrete) Sugeno integral can be defined as follows.

Definition 5. Let $m:\ 2^N \to [0,1]$ be a fuzzy measure, $\mathbf{x} = (x_1,\dots,x_n) \in [0,1]^n$. The Sugeno integral with respect to m is defined by

$$\mathbf{Su}_m(\mathbf{x}) = \max_{i \in N} \min\{x_{\sigma(i)}, m(E_{\sigma(i)})\} \tag{1}$$

where σ is a permutation on N such that $x_{\sigma(1)} \le \dots \le x_{\sigma(n)}$, $E_{\sigma(i)} = \{\sigma(i),\dots,\sigma(n)\}$ for $i = 1,\dots,n$.

Note, that the Sugeno integral is an idempotent function, which is also internal, comonotone maxitive, comonotone minitive, min-homogeneous and it extends the fuzzy measure, i.e., $\mathbf{Su}_m(\mathbb{1}_A) = m(A)$, where $\mathbb{1}_A$ is the indicator of the set $A \subseteq N$, i.e.,

$$\mathbb{1}_A(i) = \begin{cases} 1 & \text{if } i \in A \\ 0 & \text{otherwise} \end{cases},$$

for each $i \in N$.

The following generalization of the formula (1) was introduced in [25], replacing the minimum and maximum operators by some more general functions.

Definition 6. Let $m\colon 2^N \to [0,1]$ be a symmetric fuzzy measure, $F\colon [0,\infty[\times[0,1] \to [0,\infty[$, $G\colon [0,\infty[^n \to [0,\infty[$. The Sugeno-like FG-functional with respect to m is defined by

$$S^m_{FG}(\mathbf{x}) = G\left(F(x_{\sigma(1)}, m(E_{\sigma(1)})), \dots, F(x_{\sigma(n)}, m(E_{\sigma(n)}))\right), \tag{2}$$

with the same meaning of σ and $E_{\sigma(i)}$ as in Definition 5.

Note that the symmetry of the fuzzy measure ensures the Sugeno-like FG-functional to be well-defined.

2.3. Interval-valued aggregation functions

When aggregating data with some uncertainty represented by intervals, a need of interval-valued aggregation functions appears. Moreover, for using an interval-valued fuzzy integrals, there is also a need of a total order on the class of all intervals.

Let us denote by $L([0,1]) = \{X = [\underline{X}, \overline{X}] \mid 0 \le \underline{X} \le \overline{X} \le 1\}$ the set of all closed subintervals in $[0,1]$. We denote $\mathbf{0} = [0,0]$, $\mathbf{1} = [1,1]$. The standard partial order $\le_{spo}$ on $L([0,1])$ is given by:

$$[\underline{X}, \overline{X}] \le_{spo} [\underline{Y}, \overline{Y}] \ \Leftrightarrow\ \underline{X} \le \underline{Y} \ \text{ and } \ \overline{X} \le \overline{Y}.$$

Definition 7. An order $\preceq$ on $L([0,1])$ is called admissible, if
(i) $\preceq$ is a total order on $L([0,1])$, and
(ii) for all $X, Y \in L([0,1])$ it holds $X \preceq Y$ whenever $X \le_{spo} Y$.

The following construction method for admissible orders on $L([0,1])$ was suggested in [28].

Let $M_1, M_2\colon [0,1]^2 \to [0,1]$ be aggregation functions, such that for all $X, Y \in L([0,1])$ it holds

$$M_1(\underline{X}, \overline{X}) = M_1(\underline{Y}, \overline{Y}) \wedge M_2(\underline{X}, \overline{X}) = M_2(\underline{Y}, \overline{Y}) \quad \Rightarrow \quad X = Y.$$

We define the total order relation $\le_{M_1,M_2}$ as follows:

$$X \le_{M_1,M_2} Y \text{ iff } \begin{cases} M_1(\underline{X}, \overline{X}) < M_1(\underline{Y}, \overline{Y}) \text{ or} \\ M_1(\underline{X}, \overline{X}) = M_1(\underline{Y}, \overline{Y}) \text{ and } M_2(\underline{X}, \overline{X}) \le M_2(\underline{Y}, \overline{Y}). \end{cases}$$

Now, let $\alpha \in [0,1]$, $X \in L([0,1])$. Define a function $K_\alpha(X) = (1-\alpha)\underline{X} + \alpha\overline{X}$.

Let $\alpha \neq \beta \in [0,1]$. The total order relation $\le_{M_1,M_2} =: \le_{\alpha,\beta}$ corresponding to $M_1 = K_\alpha$ and $M_2 = K_\beta$ is an admissible order.

Well-known particular cases of $\le_{\alpha,\beta}$ are the following:

- Xu and Yager order $\le_{XY} = \le_{0.5,1}$:

$$[\underline{X}, \overline{X}] \le_{XY} [\underline{Y}, \overline{Y}] \ \Leftrightarrow\ \begin{cases} \underline{X} + \overline{X} < \underline{Y} + \overline{Y} \text{ or} \\ \underline{X} + \overline{X} = \underline{Y} + \overline{Y} \text{ and } \overline{X} - \underline{X} \le \overline{Y} - \underline{Y}. \end{cases}$$

- lexicographical orders $\le_{Lex1} = \le_{0,1}$, $\le_{Lex2} = \le_{1,0}$:

$$[\underline{X}, \overline{X}] \le_{Lex_1} [\underline{Y}, \overline{Y}] \ \Leftrightarrow\ \begin{cases} \underline{X} < \underline{Y} \text{ or} \\ \underline{X} = \underline{Y} \text{ and } \overline{X} \le \overline{Y} \end{cases}$$

and

$$[\underline{X}, \overline{X}] \leq_{Lex_2} [\underline{Y}, \overline{Y}] \;\Leftrightarrow\; \begin{cases} \overline{X} < \overline{Y} \text{ or} \\ \overline{X} = \overline{Y} \text{ and } \underline{X} \leq \underline{Y}. \end{cases}$$

It was shown in [28], that for a given $\alpha \in [0, 1[$ all admissible orders $\leq_{\alpha,\beta}$ with $\beta > \alpha$ coincide. This admissible order will be denoted by $\leq_{\alpha+}$. Similarly, for a given $\alpha \in]0, 1]$ all admissible orders $\leq_{\alpha,\beta}$ with $\beta < \alpha$ coincide. This admissible order will be denoted by $\leq_{\alpha-}$.

Definition 8. Let $\preceq$ be an admissible order on $L([0, 1])$. A function $m : 2^N \to L([0, 1])$ is called an interval-valued fuzzy measure w.r.t. $\preceq$, if $m(\emptyset) = \mathbf{0}$, $m(N) = \mathbf{1}$ and $m(A) \preceq m(B)$ for all $A \subseteq B \subseteq N$.

Definition 9. Let $n \geq 2$. Let $\preceq$ be an admissible order on $L([0, 1])$. An n-dimensional interval-valued (IV) aggregation function w.r.t. $\preceq$ is a mapping $M : L([0, 1])^n \to L([0, 1])$ satisfying the following properties:

- $M(\mathbf{0}, \dots, \mathbf{0}) = \mathbf{0}$;
- $M(\mathbf{1}, \dots, \mathbf{1}) = \mathbf{1}$;
- M is an increasing function in each component w.r.t. $\preceq$.

3. An interval-valued generalization of the Sugeno integral

In this Section we give our definition of the generalized interval-valued Sugeno integral, which is an interval-valued counterpart to (2). To simplify our notation we will use the lattice symbols Min $= \wedge$ and Max $= \vee$.

Definition 10. Let $\preceq$ be an admissible order on $L([0, 1])$, $m : 2^N \to L([0, 1])$ be an IV fuzzy measure w.r.t. $\preceq$ and $F : L([0, 1]) \times L([0, 1]) \to L([0, 1])$, $G : \big(L([0, 1])\big)^n \to L([0, 1])$ be functions. We say that a triplet (m, F, G) satisfies Condition (WDS) if for all $X_1, \dots, X_n \in L([0, 1])$ and all possible permutations σ_1, σ_2 on N such that $X_{\sigma_1(1)} \preceq \dots \preceq X_{\sigma_1(n)}$ and $X_{\sigma_2(1)} \preceq \dots \preceq X_{\sigma_2(n)}$ it holds:

$$G\Big(F\big(X_{\sigma_1(1)}, m(E_{\sigma_1(1)})\big), \dots, F\big(X_{\sigma_1(n)}, m(E_{\sigma_1(n)})\big)\Big) = G\Big(F\big(X_{\sigma_2(1)}, m(E_{\sigma_2(1)})\big), \dots, F\big(X_{\sigma_2(n)}, m(E_{\sigma_2(n)})\big)\Big), \quad (3)$$

where $E_{\sigma_j(i)} = \{\sigma_j(i), \dots, \sigma_j(n)\}$ for $j \in \{1, 2\}$.

Definition 11. Let n be a positive integer, $\preceq$ be an admissible order on $L([0, 1])$ and let a triplet (m, F, G) satisfies Condition (WDS). An interval-valued Sugeno-like FG-functional with respect to m is a function $\mathbf{S}_m^{F,G} : \big(L([0, 1])\big)^n \to L([0, 1])$ given by

$$\mathbf{S}_m^{F,G}(X_1, \dots, X_n) = G\Big(F\big(X_{\sigma(1)}, m(E_{\sigma(1)})\big), \dots, F\big(X_{\sigma(n)}, m(E_{\sigma(n)})\big)\Big) \quad (4)$$

for all $X_1, \dots, X_n \in L([0, 1])$, where σ is a permutation on N such that $X_{\sigma(1)} \preceq \dots \preceq X_{\sigma(n)}$ and $E_{\sigma(i)} = \{\sigma(i), \dots, \sigma(n)\}$.

The following result is immediate.

Lemma 12. *Let $g : L([0, 1]) \to L([0, 1])$ be a function. If a triplet (m, F, G) satisfies Condition (WDS), then the triplet $(m, F, g \circ G)$ satisfies Condition (WDS).*

Conditions under which a triplet (m, F, G) satisfies Condition (WDS), i.e. under which the function $\mathbf{S}_m^{F,G}$ is well defined, are given in the following proposition.

Proposition 13. *Let $F : L([0, 1]) \times L([0, 1]) \to L([0, 1])$, $G : \big(L([0, 1])\big)^n \to L([0, 1])$ be functions and $m : 2^N \to L([0, 1])$ be an IV fuzzy measure with respect to an admissible order $\preceq$. Then the following hold:*

(i) *A triplet (m, F, G) satisfies Condition (WDS) for any symmetric m.*
(ii) *Let $G = f \circ Proj_1$ for some function $f : L([0, 1]) \to L([0, 1])$. Then a triplet (m, F, G) satisfies Condition (WDS) for any m.*
(iii) *Let F be non-decreasing in the second variable and $G = f \circ \vee$ for some function $f : L([0, 1]) \to L([0, 1])$. Then a triplet (m, F, G) satisfies Condition (WDS) for any m.*

Proof. A triplet (m, F, G) satisfies Condition (WDS) if the value of the functional defined by Formula (4) is for each vector $(X_1, \dots, X_n) \in L([0, 1])$ independent of the choice of permutation ordering this vector in accordance with given admissible order $\preceq$ on $L([0, 1])$. We have to discuss the case when some ties occur in vector $(X_1, \dots, X_n)$.

(i) Let $X_1, \dots, X_n \in L([0, 1])$ and σ_1, σ_2 be permutations on N such that there is a tie:

$$X_{\sigma_1(k)} = \dots = X_{\sigma_1(p)} = X_{\sigma_2(k)} = \dots = X_{\sigma_2(p)} \tag{5}$$

for some $1 \le k < p \le n$, where $X_{\sigma_1(k-1)} \prec X_{\sigma_1(k)}$ if $k > 1$ and $X_{\sigma_1(p+1)} \succ X_{\sigma_1(p)}$ if $p < n$. Then the symmetry of measure m implies

$$\begin{gathered}F\Big(X_{\sigma_1(k+1)}, m\big(\{\sigma_1(k+1), \dots, \sigma_1(n)\}\big)\Big) = F\Big(X_{\sigma_2(k+1)}, m\big(\{\sigma_2(k+1), \dots, \sigma_2(n)\}\big)\Big)\\ \vdots \\ F\Big(X_{\sigma_1(p)}, m\big(\{\sigma_1(p), \dots, \sigma_1(n)\}\big)\Big) = F\Big(X_{\sigma_2(p)}, m\big(\{\sigma_2(p), \dots, \sigma_2(n)\}\big)\Big)\end{gathered} \tag{6}$$

hence Condition (WDS) is satisfied.
(ii) Directly follows from Equation (4).
(iii) Considering the same tie as in (i), by the monotonicity of F and m, we have:

$$\begin{aligned}&\vee\Bigg(F\Big(X_{\sigma_1(k)}, m\big(\{\sigma_1(k), \dots, \sigma_1(n)\}\big)\Big), \dots, F\Big(X_{\sigma_1(p)}, m\big(\{\sigma_1(p), \dots, \sigma_1(n)\}\big)\Big)\Bigg) =\\ &F\Big(X_{\sigma_1(k)}, m\big(\{\sigma_1(k), \dots, \sigma_1(n)\}\big)\Big) = F\Big(X_{\sigma_2(k)}, m\big(\{\sigma_2(k), \dots, \sigma_2(n)\}\big)\Big) =\\ &\vee\Bigg(F\Big(X_{\sigma_2(k)}, m\big(\{\sigma_2(k), \dots, \sigma_2(n)\}\big)\Big), \dots, F\Big(X_{\sigma_2(p)}, m\big(\{\sigma_2(p), \dots, \sigma_2(n)\}\big)\Big)\Bigg).\end{aligned} \tag{7}$$

Hence, the triplet $(m, F, \vee)$ satisfies Condition (WDS), and consequently, by Lemma 12, also triplet the (m, F, G) satisfies the condition. □

Remark 14. Observe that the three instances of well defined triplets (m, F, G) in Proposition 13 are not the only ones, but they involve all non-trivial cases. For instance, we do not mention the case of $F(X, Y) = f(X)$ for some function $f\colon L([0, 1]) \to L([0, 1])$ which together with an arbitrary function $G\colon L([0, 1])^n \to L([0, 1])$ yields a well-defined function $S_m^{F,G}$ for all fuzzy measures m. In fact, in that case $S_m^{F,G}$ does not depend on a fuzzy measure m.

Since in the definition of standard Sugeno integral the functions $G = \vee$ and $F = \wedge$ are used, we also consider the three special cases of IV Sugeno-like FG-functionals, in particular $\mathbf{S}_m^{\wedge,G}$, $\mathbf{S}_m^{F,\vee}$ and $\mathbf{S}_m^{\wedge,\vee}$.

Corollary 15. *Let $F\colon L([0, 1]) \times L([0, 1]) \to L([0, 1])$, $G\colon \big(L([0, 1])\big)^n \to L([0, 1])$ be functions and $m\colon 2^N \to L([0, 1])$ be an IV fuzzy measure with respect to an admissible order $\preceq$. Then the following hold:*

(i) *Let F be non-decreasing in the second variable. Then a triplet $(m, F, \vee)$ satisfies Condition (WDS) for any m.*
(ii) *Let $G = f \circ Proj_1$ or $G = f \circ \vee$ for some function $f : L([0, 1]) \to L([0, 1])$. Then a triplet $(m, \wedge, G)$ satisfies Condition (WDS) for any m.*
(iii) *A triplet $(m, \wedge, \vee)$ satisfies Condition (WDS) for any m.*

Proof. (i) Directly follows from Proposition 13 (iii).

(ii) Directly follows from Proposition 13 (ii)-(iii).
(iii) Follows from item (i). □

Taking $F = \vee$, $G = \wedge$ and degenerate intervals (i.e. intervals $X = [\underline{X}, \overline{X}]$ with $\underline{X} = \overline{X}$) as inputs the IV Sugeno-like FG-functional recovers standard Sugeno integral on $[0, 1]$.

Proposition 16. *Let $\mu : 2^N \to [0, 1]$ be a fuzzy measure and $m : 2^N \to L([0, 1])$ be the IV fuzzy measure given by $m(A) = [\mu(A), \mu(A)]$ for all $A \subseteq N$. Let $\mathbf{S}_m^{\wedge,\vee} : \big(L([0, 1])\big)^n \to L([0, 1])$ be an IV Sugeno-like FG-functional with respect to m for $F = \vee$ and $G = \wedge$. Then the function $f : [0, 1]^n \to [0, 1]$ given by*

$$f(x_1, \dots, x_n) = y \qquad \text{where} \qquad \mathbf{S}_m^{\wedge,\vee}([x_1, x_1], \dots, [x_n, x_n]) = [y, y], \tag{8}$$

is the Sugeno integral on $[0, 1]$ with respect to μ.

Proof. The proof follows from the observation that for any admissible order $\preceq$ and all $x, y \in [0, 1]$ it holds $[x, x] \preceq [y, y]$ if and only if $x \leq y$. □

4. Properties of the IV Sugeno-like FG-functional

In this section we study the mathematical properties of the IV Sugeno-like FG-functional: idempotency, internality, positive and min-homogeneity, comonotone maxitivity and comonotone minitivity, boundary conditions, monotonicity and property of giving back the fuzzy measure.

4.1. Idempotency

Proposition 17. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}$ is idempotent for any m whenever:*

(i) *$F(X, \mathbf{1}) = X$ for all $X \in L([0, 1])$ and $G = Proj_1$; or*
(ii) *$F(X, \mathbf{1}) = X$ for all $X \in L([0, 1])$, F is non-decreasing in the second variable and $G = \vee$; or*
(iii) *G is idempotent and $F(X, Y) = X$ for all $X, Y \in L([0, 1])$.*

Proof. By Proposition 13 the functional $\mathbf{S}_m^{F,G} : \big(L([0, 1])\big)^n \to L([0, 1])$ is well-defined in all cases (i)-(iii). The fact that $\mathbf{S}_m^{F,G}(X, \dots, X) = X$ for all $X \in L([0, 1])$ is easy to check. □

Remark 18. It is worth to put out that relaxing the assumption:

F is non-decreasing in the second variable

in Proposition 17 (ii) by the assumption:

$$F(X, Y) \preceq F(X, \mathbf{1}) \text{ for all } X, Y \in L([0, 1]) \tag{9}$$

we obtain a sufficient condition under which $\mathbf{S}_m^{F,\vee}$ is idempotent for any symmetric m.

Corollary 19.

(i) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,\vee}$ is idempotent for any m whenever $F(X, \mathbf{1}) = X$ for all $X \in L([0, 1])$ and F is non-decreasing in the second variable.*
(ii) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{\wedge,G}$ is idempotent for any m whenever $G = \vee$ or $G = Proj_1$.*
(iii) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{\wedge,\vee}$ is idempotent for any m.*

Note that, as it is in item (iii) of Proposition 17, if $F(X, Y) = X$ for all $X, Y \in L([0, 1])$, then $\mathbf{S}_m^{F,G} = G$ so the Sugeno-like FG-functional does not depend on fuzzy measure m and we obtain a trivial case.

4.2. Internality

Recall that a function is called internal if it is between minimum and maximum. We study the conditions under which an interval-valued Sugeno-like FG-functional is internal for any (symmetric) fuzzy measure.

Proposition 20. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}$ satisfies:*

(i) $\mathbf{S}_m^{F,G} \succeq \wedge$ *for any m whenever*
  (a) $F(X, \mathbf{1}) \succeq X$ *for all* $X \in L([0,1])$ *and* $G = f \circ Proj_1$ *for some function* $f : L([0,1]) \to L([0,1])$ *such that* $f \succeq Id$*; or*
  (b) $F(X, \mathbf{1}) \succeq X$ *for all* $X \in L([0,1])$*, F is non-decreasing in the second variable and* $G = f \circ \vee$ *for some function* $f : L([0,1]) \to L([0,1])$ *such that* $f \succeq Id$.

(ii) $\mathbf{S}_m^{F,G} \preceq \vee$ *for any m whenever*
  (a) $F(X, \mathbf{1}) \preceq X$ *for all* $X \in L([0,1])$ *and* $G = f \circ Proj_1$ *for some function* $f : L([0,1]) \to L([0,1])$ *such that* $f \preceq Id$*; or*
  (b) $F(X, \mathbf{1}) \preceq X$ *for all* $X \in L([0,1])$*, F is non-decreasing in the second variable and* $G = f \circ \vee$ *for some function* $f : L([0,1]) \to L([0,1])$ *such that* $f \preceq Id$.

(iii) $\mathbf{S}_m^{F,G}$ *is internal for any m whenever*
  (a) $F(X, \mathbf{1}) = X$ *for all* $X \in L([0,1])$ *and* $G = Proj_1$*; or*
  (b) $F(X, \mathbf{1}) = X$ *for all* $X \in L([0,1])$*, F is non-decreasing in the second variable and* $G = \vee$.

Proof. By Proposition 13 and Corollary 15 the functional $\mathbf{S}_m^{F,G} : \big(L([0,1])\big)^n \to L([0,1])$ is well-defined in all cases. The fact that $\mathbf{S}_m^{F,G}$ is internal is easy to check in each case separately. □

Remark 21. It is worth to put out that relaxing the assumption:

$$F \text{ is non-decreasing in the second variable}$$

in Proposition 20 (i)(b) (or Proposition 20 (ii)(b), or Proposition 20 (iii)(b)) by the assumption:

$$F(X, Y) \preceq F(X, \mathbf{1}) \quad \text{for all } X, Y \in L([0,1]) \tag{10}$$

we obtain a sufficient condition under which $\mathbf{S}_m^{F,G} \succeq \wedge$ (or $\mathbf{S}_m^{F,G} \preceq \vee$, or $\mathbf{S}_m^{F,G}$ is internal) for any symmetric m.

Corollary 22.

(i) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,\vee}$ satisfies:*
  (a) $\mathbf{S}_m^{F,\vee} \succeq \wedge$ *for any m whenever* $F(X, \mathbf{1}) \succeq X$ *for all* $X \in L([0,1])$ *and F is non-decreasing in the second variable.*
  (a^s) $\mathbf{S}_m^{F,\vee} \succeq \wedge$ *for any symmetric m whenever* $F(X, \mathbf{1}) \succeq X$ *for all* $X \in L([0,1])$.
  (b) $\mathbf{S}_m^{F,\vee} \preceq \vee$ *for any m whenever* $F(X, \mathbf{1}) \preceq X$ *for all* $X \in L([0,1])$ *and F is non-decreasing in the second variable.*
  (b^s) $\mathbf{S}_m^{F,\vee} \preceq \vee$ *for any symmetric m whenever* $F(X, Y) \preceq X$ *for all* $X, Y \in L([0,1])$.
  (c) $\mathbf{S}_m^{F,\vee}$ *is internal for any m whenever* $F(X, \mathbf{1}) = X$ *for all* $X \in L([0,1])$ *and F is non-decreasing in the second variable.*
  (c^s) $\mathbf{S}_m^{F,\vee}$ *is internal for any symmetric m whenever* $F(X, Y) \preceq X$ *and* $F(X, \mathbf{1}) = X$ *for all* $X \in L([0,1])$.

(ii) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{\wedge,G}$ satisfies:*
  (a) $\mathbf{S}_m^{\wedge,G} \succeq \wedge$ *for any m whenever* $G = f \circ Proj_1$ *or* $G = f \circ \vee$ *for some function* $f\colon L([0,1]) \to L([0,1])$ *such that* $f \succeq Id$.
  (b) $\mathbf{S}_m^{\wedge,G} \preceq \vee$ *for any m whenever* $G = f \circ Proj_1$ *or* $G = f \circ \vee$ *for some function* $f\colon L([0,1]) \to L([0,1])$ *such that* $f \preceq Id$.
  (c) $\mathbf{S}_m^{\wedge,G}$ *is internal for any m whenever* $G = Proj_1$ *or* $G = \vee$.

(iii) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{\wedge,\vee}$ is internal for any m.*

4.3. Positive and min-homogeneity

Recall that a function $f: L([0,1])^n \to L([0,1])$ is called positively homogeneous if $f(cX_1,\dots,cX_n) = cf(X_1,\dots,X_n)$ for all $(X_1,\dots,X_n) \in L([0,1])^n$ and all $c \in \mathbb{R}^+$ such that $c(X_1,\dots,X_n) = (cX_1,\dots,cX_n) \in L([0,1])^n$ with convention $c[\underline{X},\overline{X}] = [c\underline{X}, c\overline{X}]$.

Similarly, a function $f: L([0,1])^n \to L([0,1])$ is called min-homogeneous if $f(c \wedge (X_1,\dots,X_n)) = c \wedge f(X_1,\dots,X_n)$ for all $(X_1,\dots,X_n) \in L([0,1])^n$ and all $c \in \mathbb{R}^+$ such that $c \wedge (X_1,\dots,X_n) = (c \wedge X_1,\dots,c \wedge X_n) \in L([0,1])^n$, with convention $c \wedge [\underline{X},\overline{X}] = [c \wedge \underline{X}, c \wedge \overline{X}]$. We study the conditions under which an interval-valued Sugeno-like FG-functional is positively (min-)homogeneous for any (symmetric) fuzzy measure.

Proposition 23. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}$ is positively (min-)homogeneous for any m whenever G is positively (min-)homogeneous and $F(\cdot,y)$ is positively (min-)homogeneous for every $y \in L([0,1])$.*

Proof. Easy to check. □

Corollary 24.

(i) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,\vee}$ is positively (min-)homogeneous for any m whenever $F(\cdot,y)$ is positively (min-)homogeneous for every $y \in L([0,1])$.*
(ii) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{\wedge,G}$ is min-homogeneous for any m whenever G is min-homogeneous.*
(iii) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{\wedge,\vee}$ is min-homogeneous for any m.*

4.4. Comonotone maxitivity

Recall that a function $f: L([0,1])^n \to L([0,1])$ is called comonotone maxitive if $f(X \vee Y)) = f(X) \vee f(Y)$ for all $X = (X_1,\dots,X_n) \in L([0,1])^n$ and $Y = (Y_1,\dots,Y_n) \in L([0,1])^n$ which are comonotone, i.e., there exists a permutation σ on N with $X_{\sigma(1)} \preceq \dots \preceq X_{\sigma(n)}$ and $Y_{\sigma(1)} \preceq \dots \preceq Y_{\sigma(n)}$.

Proposition 25. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}$ is comonotone maxitive for any m whenever G is comonotone maxitive and $F(\cdot,y)$ is comonotone maxitive for every $y \in L([0,1])$.*

Proof. Easy to check. □

Corollary 26.

(i) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,\vee}$ is comonotone maxitive for any m whenever $F(\cdot,y)$ is comonotone maxitive for every $y \in L([0,1])$.*
(ii) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{\wedge,G}$ is comonotone maxitive for any m whenever G is comonotone maxitive.*
(iii) *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{\wedge,\vee}$ is comonotone maxitive for any m.*

4.5. Boundary conditions and monotonicity

Proposition 27. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}: \big(L([0,1])\big)^n \to L([0,1])$ fulfills condition $\mathbf{S}_m^{F,G}(\mathbf{0},\dots,\mathbf{0}) = \mathbf{0}$ for any m whenever:*

(i) *$F(\mathbf{0},\mathbf{1}) = \mathbf{0}$, $G = f \circ Proj_1$ and $f(\mathbf{0}) = \mathbf{0}$; or*
(ii) *$F(\mathbf{0},Y) = \mathbf{0}$ for all $Y \in L([0,1])$ and $G = f \circ \vee$ and $f(\mathbf{0}) = \mathbf{0}$*

Proof. By Proposition 13 the functional $\mathbf{S}_m^{F,G}$ is well-defined in cases (i)-(ii). The fact that $\mathbf{S}_m^{F,G}(\mathbf{0},\dots,\mathbf{0}) = \mathbf{0}$ is easy to check. □

Proposition 28. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}\colon \big(L([0,1])\big)^n \to L([0,1])$ fulfills condition $\mathbf{S}_m^{F,G}(\mathbf{0},\dots,\mathbf{0}) = \mathbf{0}$ for any symmetric m whenever $G(\mathbf{0},\dots,\mathbf{0}) = \mathbf{0}$ and $F(\mathbf{0},Y) = \mathbf{0}$ for all $Y \in L([0,1])$.*

Proposition 29. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}\colon \big(L([0,1])\big)^n \to L([0,1])$ fulfills condition $\mathbf{S}_m^{F,G}(\mathbf{1},\dots,\mathbf{1}) = \mathbf{1}$ for any m whenever:*

(i) *$F(\mathbf{1},\mathbf{1}) = \mathbf{1}$, $G = f \circ Proj_1$ and $f(\mathbf{1}) = \mathbf{1}$; or*
(ii) *$F(\mathbf{1},\mathbf{1}) = \mathbf{1}$ and F is non-decreasing in the second variable, $G = f \circ \vee$ and $f(\mathbf{1}) = \mathbf{1}$.*

Proposition 30. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}\colon \big(L([0,1])\big)^n \to L([0,1])$ fulfills condition $\mathbf{S}_m^{F,G}(\mathbf{1},\dots,\mathbf{1}) = \mathbf{1}$ for any symmetric m whenever $G(\mathbf{1},\dots,\mathbf{1}) = \mathbf{1}$ and $F(\mathbf{1},Y) = \mathbf{1}$ for all $Y \in L([0,1])$.*

Proposition 31. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}\colon \big(L([0,1])\big)^n \to L([0,1])$ is non-decreasing for any m whenever:*

(i) *$F(\cdot,\mathbf{1})$ is non-decreasing, $G = f \circ Proj_1$ and f is non-decreasing; or*
(ii) *F is non-decreasing, $G = f \circ \vee$ and f is non-decreasing.*

Proposition 32. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}\colon \big(L([0,1])\big)^n \to L([0,1])$ is non-decreasing for any symmetric m whenever G is non-decreasing and F is non-decreasing in the first variable.*

Proof. By Proposition 13 the functional $\mathbf{S}_m^{F,G}$ is well-defined. The claim is easy to check. □

Corollary 33. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}$ is an aggregation function for any m whenever*

(i) *$F(\cdot,\mathbf{1})$ is non-decreasing, $F(\mathbf{0},\mathbf{1}) = \mathbf{0}$, $F(\mathbf{1},\mathbf{1}) = \mathbf{1}$, $G = f \circ Proj_1$ and f is non-decreasing with $f(\mathbf{0}) = \mathbf{0}$, $f(\mathbf{1}) = \mathbf{1}$; or*
(ii) *$F(\mathbf{0},Y) = \mathbf{0}$ for all $Y \in L([0,1])$, $F(\mathbf{1},\mathbf{1}) = \mathbf{1}$ and F is non-decreasing, and $G = f \circ \vee$ and f is non-decreasing with $f(\mathbf{0}) = \mathbf{0}$, $f(\mathbf{1}) = \mathbf{1}$.*

Corollary 34. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}$ is an aggregation function for any symmetric m whenever G is non-decreasing with $G(\mathbf{0},\dots,\mathbf{0}) = \mathbf{0}$, $G(\mathbf{1},\dots,\mathbf{1}) = \mathbf{1}$ and F is non-decreasing in the first variable with $F(\mathbf{0},Y) = \mathbf{0}$ and $F(\mathbf{1},Y) = \mathbf{1}$ for all $Y \in L([0,1])$.*

Example 4.1. Let $G(X_1,\dots,X_n) = \frac{1}{n}\sum_{i=1}^n X_i$; $F(X,Y) = X^2 \cdot Y + X \cdot (1-Y)$ and $m(A) = \left(\frac{|A|}{n}\right)^2$ for $A \subseteq N$. Then $S_m^{F,G}(X_1,\dots,X_n) = \frac{1}{n}\sum_{i=1}^n X_i + \frac{1}{n}\sum_{i=1}^n \left(\frac{n-i}{n}\right)^2 \left(X_i^2 - X_i\right)$ which is an aggregation function according to Corollary 34.

4.6. Giving back the fuzzy measure

Recall that one of the important properties of the standard Sugeno integral is that it gives back the fuzzy measure, i.e. $\mathbf{Su}_m(\mathbb{1}_E) = m(E)$ for all $E \subseteq N$, where $\mathbb{1}_E$ is the indicator of the set E. A modification of this property for an interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,G}$ can be formulated as follows:

$\mathbf{S}_m^{F,G}(\mathbb{1}_E) = F(\mathbf{1}, m(E))$ for all $E \subseteq N$, where $\mathbb{1}_E \in L([0,1])^n$ is the interval-valued indicator of the set E defined by $\mathbb{1}_E(i) = \mathbf{1}$ if $i \in E$ and $\mathbb{1}_E(i) = \mathbf{0}$ otherwise.

Obviously, this property cannot be satisfied in the case of $G = f \circ Proj_1$ nor in the case when we consider only symmetric measures and general G's. See Table 1.

Proposition 35. *An interval-valued Sugeno-like FG-functional $\mathbf{S}_m^{F,\vee}$ gives back the fuzzy measure for any m whenever $F(\mathbf{0},\mathbf{1}) = \mathbf{0}$ and F is non-decreasing in the second variable.*

Table 1
Summary of sufficient conditions for particular properties of the Sugeno-like FG-functionals in three cases of Proposition 13. Symbol $\nearrow$ denotes non-decreasingness.

Property	$G = f \circ \vee$, $F \nearrow$ in sec.var.	$G = f \circ Proj_1$	m symmetric
$\mathbf{S}_m^{F,G}(\mathbf{0},\ldots,\mathbf{0}) = \mathbf{0}$	$F(\mathbf{0},\cdot) = \mathbf{0}$, $f(\mathbf{0}) = \mathbf{0}$	$F(\mathbf{0},\mathbf{1}) = \mathbf{0}$, $f(\mathbf{0}) = \mathbf{0}$	$F(\mathbf{0},\cdot) = \mathbf{0}$, $G(\mathbf{0},\ldots,\mathbf{0}) = \mathbf{0}$
$\mathbf{S}_m^{F,G}(\mathbf{1},\ldots,\mathbf{1}) = \mathbf{1}$	$F(\mathbf{1},\mathbf{1}) = \mathbf{1}$, $f(\mathbf{1}) = \mathbf{1}$	$F(\mathbf{1},\mathbf{1}) = \mathbf{1}$, $f(\mathbf{1}) = \mathbf{1}$	$F(\mathbf{1},\cdot) = \mathbf{1}$, $G(\mathbf{1},\ldots,\mathbf{1}) = \mathbf{1}$
Non-decreasingness	$f \nearrow$	$f \nearrow$	$G \nearrow$, $F \nearrow$ in first var.
Aggregation function	the above lines in this table	the above lines in this table	the above lines in this table
Idempotency	$f = \mathrm{id}$	$f = \mathrm{id}$	G idempotent, $F(X,Y) = X$
Internality	$f = \mathrm{id}$, $F(X,\mathbf{1}) = X$	$f = \mathrm{id}$, $F(X,\mathbf{1}) = X$	–
Pos. homogeneity	f pos. hom., F pos. hom.	f pos. hom., F pos. hom.	G pos. hom., F pos. hom.
Min-homogeneity	f min-hom., F min-hom.	G min-hom., F min-hom.	G min-hom., F min-hom.
Comonotone maxitivity	F com. maxitive	F com. maxitive	G com.max., F com. max.
Giving back the fuzzy measure	$f = \mathrm{id}$, $F(\mathbf{0},\mathbf{1}) = \mathbf{0}$	–	–

Proof. Let E be subset of N with $\mathrm{card}(E) = k$. Then $(\mathbb{1}_E)_{\sigma(i)} = \mathbf{0}$ for $i = 1,\ldots,n-k$, $(\mathbb{1}_E)_{\sigma(i)} = \mathbf{1}$ for $i = n-k+1,\ldots,n$ and $E_{\sigma(n-k+1)} = E$, where σ is a permutation ordering the vector $\mathbb{1}_E$ non-decreasingly. Taking into account non-decreasingness of F in the second variable, we get

$$\mathbf{S}_m^{F,\vee}(\mathbb{1}_E) = \bigvee_{i=1}^{n-k} F(\mathbf{0}, m(E_{\sigma(i)})) \vee \bigvee_{i=n-k+1}^{n} F(\mathbf{1}, m(E_{\sigma(i)})) = F(\mathbf{0},\mathbf{1}) \vee F(\mathbf{1}, m(E)), \quad (11)$$

and the claim follows. □

5. Construction of IV Sugeno-like FG-functional

In this section we construct IV Sugeno-like FG-functionals with respect to the orders $\leq_{\alpha+}$ and $\leq_{\alpha-}$. Since the width of the input intervals can be regarded as the measure of data uncertainty represented by them, it is desirable to take it into account in our construction. We use the similar approach as proposed in [29].

Definition 36 *([29]).* Let $c \in [0,1]$, $\alpha \in [0,1]$ and $Z = [x,y] \in L([0,1])$. We denote by $w(Z) = y - x$ and by $d_\alpha(c)$ the maximal possible width of an interval $X \in L([0,1])$ such that $K_\alpha(X) = c$. Moreover, we define

$$\lambda_\alpha(Z) = \frac{w(Z)}{d_\alpha(K_\alpha(Z))} \quad (12)$$

where we set $\frac{0}{0} = 1$.

Proposition 37 *([29]). For all $\alpha \in [0,1]$ and $X \in L([0,1])$ it holds that*

$$d_\alpha(K_\alpha(X)) = \wedge\left(\frac{K_\alpha(X)}{\alpha}, \frac{1-K_\alpha(X)}{1-\alpha}\right), \quad (13)$$

where we set $\frac{r}{0} = 1$ for all $r \in [0,1]$.

Let $\alpha \in [0,1]$ and let $M_1, M_2 : [0,1]^n \to [0,1]$ be n-ary functions. We define an interval function $M_{IV} : (L([0,1]))^n \to L([0,1])$ as follows:

$$M_{IV}(X_1,\ldots,X_n) = Y, \quad \text{where} \quad \begin{cases} K_\alpha(Y) = M_1\left(K_\alpha(X_1),\ldots,K_\alpha(X_n)\right), \\ \lambda_\alpha(Y) = M_2\left(\lambda_\alpha(X_1),\ldots,\lambda_\alpha(X_n)\right), \end{cases} \quad (14)$$

for all $X_1,\ldots,X_n \in L([0,1])$. It was proved in [29] that M_{IV} is an IV aggregation function with respect to $\leq_{\alpha+}$ and $\leq_{\alpha-}$ whenever M_1, M_2 be aggregation functions where M_1 is strictly increasing. Now we show that the obtained M_{IV} preserves some properties of the functions M_1 and M_2, in particular the properties of t-norms.

Proposition 38. *Let $\alpha \in [0,1]$, let $M_1, M_2 : [0,1]^n \to [0,1]$ be functions and let $M_{IV} : (L([0,1]))^n \to L([0,1])$ be an interval function given by Equation (14). Then the following hold:*

(i) $M_{IV}(\mathbf{0},\dots,\mathbf{0}) = \mathbf{0}$ *whenever* $M_1(0,\dots,0) = 0$.
(ii) $M_{IV}(\mathbf{1},\dots,\mathbf{1}) = \mathbf{1}$ *whenever* $M_1(1,\dots,1) = 1$.
(iii) *For* $n = 2$ $M_{IV}(\mathbf{0}, Y) = \mathbf{0}$ *for all* $Y \in L([0,1])$ *whenever* $M_1(0,y) = 0$ *for all* $y \in [0,1]$.
(iv) *For* $n = 2$ $M_{IV}(\mathbf{0}, Y) = Y$ *for all* $Y \in L([0,1])$ *whenever* $M_1(0,y) = y$ *and* $M_2(1,y) = y$ *for all* $y \in [0,1]$.
(v) *For* $n = 2$ $M_{IV}(\mathbf{1}, Y) = Y$ *for all* $Y \in L([0,1])$ *whenever* $M_1(1,y) = y$ *for all* $y \in [0,1]$.
(vi) M_{IV} *is symmetric whenever* M_1 *and* M_2 *are symmetric.*
(vii) M_{IV} *is associative whenever* M_1 *and* M_2 *are associative.*
(viii) M_{IV} *is non-decreasing whenever* M_1, M_2 *are non-decreasing and* M_1 *is strictly increasing on*

$$\{(x_1,\dots,x_n) \in [0,1]^n \mid M_1(x_1,\dots,x_n) \notin \{0,1\}\}. \tag{15}$$

(ix) *Let* $k \in N$, *then* M_{IV} *is non-decreasing in the* k*-th variable whenever* M_1, M_2 *are non-decreasing in the* k*-th variable and* M_1 *is strictly increasing in the* k*-th variable on*

$$\{(x_1,\dots,x_n) \in [0,1]^n \mid M_1(x_1,\dots,x_n) \notin \{0,1\}\}. \tag{16}$$

Proof. The items (i), (ii) and (viii) follow from [28, Theorem 3.16]. With respect to (iv) observe:

$$K_\alpha\big(M_{IV}(\mathbf{0}, Y)\big) = M_1\big(K_\alpha(\mathbf{0}), K_\alpha(Y)\big) = M_1\big(0, K_\alpha(Y)\big) = K_\alpha(Y) \tag{17}$$

and

$$\lambda_\alpha\big(M_{IV}(\mathbf{0}, Y)\big) = M_2\big(\lambda_\alpha(\mathbf{0}), \lambda_\alpha(Y)\big) = M_1\big(1, \lambda_\alpha(Y)\big) = \lambda_\alpha(Y). \tag{18}$$

The proofs of (iii) and (v) are similar to that of (iv) observing that if $K_\alpha(Y) = 0$ (or $K_\alpha(Y) = 1$), then $Y = \mathbf{0}$ (or $Y = \mathbf{1}$) regardless the value of $\lambda_\alpha(Y)$. The items (vi) and (ix) are straightforward. Finally, with respect to (vii), let $U = M_{IV}\big(M_{IV}(X,Y), Z\big)$ and $V = M_{IV}\big(X, M_{IV}(Y,Z)\big)$. Then, by the associativity of M_1, we have

$$K_\alpha(U) = M_1\Big(M_1\big(K_\alpha(X), K_\alpha(Y)\big), K_\alpha(Z)\Big) = M_1\Big(K_\alpha(X), M_1\big(K_\alpha(Y), K_\alpha(Z)\big)\Big) = K_\alpha(V) \tag{19}$$

and similarly, by the associativity of M_2, also $\lambda_\alpha(U) = \lambda_\alpha(V)$. □

Remark 39. It is worth to put out that, taking $M_1 = \wedge$ (i.e., minimum in [0, 1]), the induced M_{IV} is not minimum in $L([0,1])$ for any M_2. In particular, it is easy to check that since minimum in [0, 1] is not strictly increasing, the monotonicity of M_{IV} is violated. Hence, $\mathbf{S}_m^{\wedge,\vee} \neq \mathbf{S}_m^{M_{IV},\vee}$ if M_{IV} is induced by $M_1 = \wedge$.

Corollary 40. *Under the assumptions of Proposition 38, if* $n = 2$ *then the following hold:*

(i) M_{IV} *is an IV* t*-norm whenever* M_1 *is a strictly increasing* t*-norm and* M_2 *is a commutative, associative and non-decreasing function.*
(ii) M_{IV} *is an IV* t*-conorm whenever* M_1 *is a strictly increasing* t*-conorm and* M_2 *is* t*-norm.*

Example 5.1. We give examples of IV Sugeno-like FG-functionals $\mathbf{S}_m^{F,G}$ obtained by the construction based on Proposition 13, Equation (14) and Proposition 38. The measure m need not be symmetric. The functional $\mathbf{S}_m^{F,G}$ is well-defined if we take $G = f \circ \vee$ where, for instance, $f(X) = X$ or $f(X) = X^2 = \left[\underline{X}^2, \overline{X}^2\right]$ or $f(X) = \sqrt{X} = \left[\sqrt{\underline{X}}, \sqrt{\overline{X}}\right]$; and F induced by M_1, M_2: $[0,1]^2 \to [0,1]$ where, for instance:

(i) $M_1(x,y) = xy$ and $M_2 = M_1$.
(ii) $M_1(x,y) = xy$ and $M_2(x,y) = (1-a_2)x + a_2 y$ for some $a_2 \in [0,1]$.
(iii) $M_1(x,y) = \begin{cases} 0, & \text{if } x = y = 0 \\ \frac{xy}{x+y-xy}, & \text{otherwise} \end{cases}$ and $M_2 = M_1$.

Table 2
In the second column, the list of the Sugeno-like FG-functionals constructed in Example 5.1 satisfying the property in the first column is given. In the third column, the justification is indicated.

Property	Sugeno-like FG-functionals	Justification
Well-defined	(i), (ii), (iii), (iv), (v)	Proposition 13 (iii)
$\mathbf{S}_m^{F,G}(\mathbf{0},\dots,\mathbf{0})=\mathbf{0}$	(i), (ii), (iii), (iv)	Proposition 38 (iii)
$\mathbf{S}_m^{F,G}(\mathbf{1},\dots,\mathbf{1})=\mathbf{1}$	(i), (ii), (iii), (iv), (v)	Proposition 38 (ii)
Non-decreasingness	(i), (ii), (iii), (iv), (v)	Proposition 38 (viii)
Aggregation function	(i), (ii), (iii), (iv)	The above lines of this table
Idempotency	(i), (iii), only for $G=\vee$	Proposition 17 (i)
Internality	(i), (iii), only for $G=\vee$	Proposition 20 (iii) (b)

(iv) $M_1(x,y)=\begin{cases}0, & \text{if } x=y=0\\ \frac{xy}{x+y-xy}, & \text{otherwise}\end{cases}$ and $M_2(x,y)=(1-a_2)x+a_2y$ for some $a_2\in[0,1]$.

(v) $M_1(x,y)=(1-a_1)x+a_1y$ and $M_2(x,y)=(1-a_2)x+a_2y$ for some $a_1\in]0,1[$ and $a_2\in[0,1]$.

Note that:

- In all the relevant cases, (ii), (iv) and (v), by the parameters a_1,a_2 we can regulate the relative weight we put on inputs $X_{\sigma(i)}$ and fuzzy measures $m(E_{\sigma(i)})$, in particular, taking $a_1=0.5$ we put the same weight to $X_{\sigma(i)}$ as to $m(E_{\sigma(i)})$; taking $a_1=0.25$ we put the greater weight to $X_{\sigma(i)}$ as to $m(E_{\sigma(i)})$ and vice versa for $a_1=0.75$.
- The measure m need not to be symmetric in the above examples (i)-(v). For symmetric measures, there are no restrictions for F and G, see Proposition 13 (i), so we can apply any functions $F: L([0,1])\times L([0,1])\to L([0,1])$, $G:\big(L([0,1])\big)^n\to L([0,1])$.
- The function M_1 in items (iii)-(iv) is the Hamacher product so it is possible to generalize the class of examples by taking $M_1(x,y)=\frac{xy}{\gamma+(1-\gamma)(x+y-xy)}$ for any $\gamma\geq 0$. In fact, we can apply as M_1 any strict t-norm.
- In Table 2 the summary of the properties satisfied by the Sugeno-like FG-functionals constructed in the above examples (i)-(v) is given.

6. IV Sugeno-like FG-functional applied in a brain computer interface

In this section, we use the proposed IV Sugeno-like FG-functional in a BCI framework that uses interval-valued predictions. We detail this framework, how the interval-valued logits are constructed, and how different versions the proposed IV-Sugeno compare to other IV-aggregations, and to other BCI frameworks.

6.1. Motor imagery brain computer interface framework

BCI systems usually consist of four different modules. The initial step is the EEG data acquisition and conditioning (signal amplification and different filters to remove noise, high impedance sensors, etc.). The second block of procedures is related to extracting features. It often includes filtering the EEG signal in one or several frequency bands, which can be subject-specific or fixed for all participants [30,31]. This block might include some dimensionality reduction procedure such as Spatio-spectral decomposition (SSD) [32,33]. Another common step is the application of an optimized spatial filtering method such as common spatial patterns (CSP, [34–38]). In case of focusing on power-based features, they are usually gaussianized by applying the natural logarithm. Then, the classification step of the features is carried out, usually based on linear classifiers ([39–41]). Often classifier ensembles are implemented, where the final decision is performed combining the outputs of all classifiers [42,12]. The last module is related to the feedback to the user, which usually is in form of visual feedback, but can also be auditory or somatosensory [43–45].

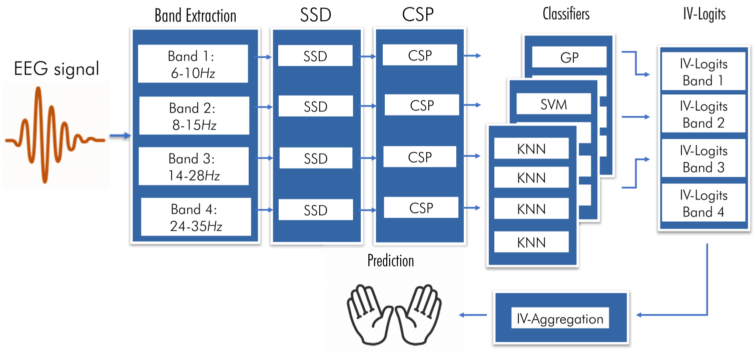


Fig. 1. Graphical depiction of the BCI framework. After measuring the EEG, SSD is computed in four different bands to reduce dimensions and increase the signal-to-noise ratio. Then CSP is applied.

6.2. Feature extraction and classification

Features from the EEG were extracted from four subject-independent and overlapped frequency bands which cover the range from low alpha to high beta in the following ranges: 6-10, 8-15, 14-28 and 24-35 Hz. The time interval where the features were extracted was optimized for each band and class-pair by analyzing the event-related desynchronization/synchronization (ERD/ERS) time courses over the sensorimotor channels [46,47]. For that, the envelope of the EEG signals was estimated with the magnitude of the analytic band pass filtered EEG and averaged over trials of each class separately. The final time interval was chosen adding those time samples whose the pair-wise discriminability using the envelopes as features were higher. The selected time intervals were used to crop the EEG signals and then SSD was computed to reduce the number of dimensions in a specific band [32,33]. SSD decomposes multivariate data into sources of maximal signal-to-noise ratio (SNR) in a narrow-band. The selected sources with high SNR were then projected on to a few common spatial directions [46,48]. The power of the projected training signals was then computed for each trial and the natural logarithm applied. Finally, LDA classifiers were trained for subsequent classification, see Fig. 1.

Regarding the testing set, the features were extracted as follows: the EEG was filtered in the four narrow bands of interest and then projected onto the corresponding SSD and CSP filters (depending on the class pair and band). The data were cropped in the time intervals selected with the training set. Finally, the power of each trial and projection was estimated and the natural logarithm applied. These features were then classified using the LDA previously trained.

6.3. Interval-valued predictions using different classifiers

In order to generate interval-valued outputs from our system, we have opted to use the predictions from different classifiers. We use the variability between them in order to estimate the uncertainty related to each of our predictions. The process is the following:

1. We choose a set of n different kinds of classifiers. For our experimentation, we have chosen three different types: Support Vector Machine (SVM), Gaussian Process (GP) and K-Nearest Neighbors (KNN).
2. From the features obtained from each of the wave bands studied, we train a classifier of each kind. In our case, this means that we train a SVM, a GP and a KNN for each wave band.
3. For each wave band, we have n different predictions for each different class. We generate an interval for each wave band and class using the lowest and the maximum predictions respectively.

After the final step, we have obtained for each class one interval-valued prediction per wave band. Then, we aggregate the interval-valued predictions for each sample using a suitable aggregation function and a $\preceq_{\alpha,\beta}$ order. Finally, we choose the maximum value according to the previously established order.

Table 3
Results for the CBCIC dataset using different interval-valued aggregations.

Aggregation function	Accuracy	F1-Score
IV-Sugeno$_1$	**0.8175 ± 0.1342**	**0.8149 ± 0.1366**
IV-Sugeno$_2$	0.8162 ± 0.1338	0.8138 ± 0.1359
IV-Sugeno$_3$	0.7994 ± 0.1324	0.8014 ± 0.1313
IV-OWA$_1$	0.8141 ± 0.1327	0.8127 ± 0.1340
IV-OWA$_2$	0.8162 ± 0.1338	0.8138 ± 0.1359
IV-OWA$_3$	0.8100 ± 0.1327	0.8073 ± 0.1352
IV-MD	0.8097 ± 0.1318	0.8085 ± 0.1315

Table 4
Results for the CBCIC dataset using different BCI frameworks.

Framework	Accuracy	F1-Score
IV-Sugeno$_1$	**0.8175 ± 0.1342**	**0.8149 ± 0.1366**
EEG Net [50]	0.6562 ± 0.1232	0.5933 ± 0.1712
Shallow Net [51]	0.7453 ± 0.13289	0.7342 ± 0.1489
Deep Net [51]	0.5331 ± 0.1356	0.4218 ± 0.1282
Multiscale CSP [52]	0.7956 ± 0.1144	0.7911 ± 0.1175
Gradient Boosting [53]	0.5956 ± 0.1203	0.5354 ± 0.1169

6.4. Experimental results for a left/right hand BCI task

We have performed our experimentation in the Clinical BCI Challenge WCCI 2020 dataset (CBCIC) [49]. This dataset consists of brain imaging signals from 10 hemiparetic stroke patients with hand functional disability in a rehabilitation task. The data contains 80 different trials of left/right hand movements. Decoding motor cortical signals of brain-injured presents several challenges as the presence of irregular because of the altered neurodynamics.

We have tried different versions of the newly proposed IV Sugeno-like FG-functionals, using the cardinality fuzzy measure in all of them, i.e., $m(A) = \frac{|A|}{n}$ for $A \subseteq N$:

- IV-Sugeno$_1$: $G(X_1, \dots, X_n) = \frac{1}{n}\sum_{i=1}^{n} X_i$, $F(X, Y) = X^2Y + X(1 - Y)$;
- IV-Sugeno$_2$: $G(X_1, \dots, X_n) = \frac{1}{n}\sum_{i=1}^{n} X_i$, $F(X, Y) = X(1 - Y)$;
- IV-Sugeno$_3$: $G(X_1, \dots, X_n) = \max(X_1, \dots, X_n)$, $F(X, Y) = \min(X, Y)$,

for $X_1, \dots, X_n, X, Y \in L([0, 1])$.

We have also studied other interval-valued aggregation functions: the interval-valued OWA operators and the interval-valued Moderate deviation functions. We have also compared with other BCI frameworks based on Deep Learning: the EEG net [50], and two other different Deep Learning architectures [51]; a different version of CSP using different temporal scales to extract features [52]; and Gradient Boosting [53].

In order to evaluate the performance of each different proposal in this dataset, each participant's dataset was randomly sampled in ten different partitions (each with 50% train and 50% test trials), resulting in a total of 80. The final performance of each configuration was obtained averaging each single dataset accuracy and F1-Score.

The results were obtained using different aggregation functions in the decision making phase and compared the newly proposed MCAs. Both the adaptive and the non-adaptive mixing parameter were employed with a set of standard aggregations and also with the already existing penalty-based aggregation functions.

Table 3 shows the results for each of the different interval-valued aggregations used in this BCI framework. IV-Sugeno$_1$ obtained the best result, followed by IV-Sugeno$_2$ and IV-OWA$_2$. Table 4 shows the comparison between the best interval-valued aggregation using our BCI framework and other BCI frameworks. The IV-Sugeno obtained the best results in this comparison, followed by the Multiscale CSP.

7. IV Sugeno-like FG-functional applied in social network analysis

In this section we propose an interval-valued version of the affinity functions proposed in [26] to characterize the relationship between two actors in a network.

We start by recalling the notions of centrality measure and affinity function in social network analysis. Then we show how we construct the iv-affinity functions and propose new centrality measures based on them. These metrics characterize each actor based on the difference in commitment that has in its relationships.

7.1. Centrality measures in social network analysis

In social network analysis, centrality measures are metrics to ponder how relevant each node is in a structure [54,55]. Some very well known centrality measures are:

- Degree centrality: the number of edges incident upon an actor. In the case of directed networks, the degree is the sum of the number of edges incident to the actor (in-degree) and the number of edges salient to the actor (out-degree).
- Betweenness centrality: the betweenness of an actor is the number of times that node is in the shortest path of other two nodes. It measures the importance of the actor in the network's information flow.
- Closeness centrality: the closeness centrality of an actor is the average length of the shortest path between that node and the rest of the nodes in the network. It establishes a center-periphery difference.
- Eigenvector centrality: it assigns a relative score to each actor in the network based on the idea that connections to well connected actors should ponder more than connections to poorly connected ones.

7.2. Affinity functions in social network analysis

Affinity functions were proposed in [26] as a way to measure the relationship between a pair of actors in a social network using their local information. We denote by V the set of all actors. "Affinities" are defined as functions over the set V^2 of all pairs of actors in a given social network assigning a number $F_C(x, y) \in [0, 1]$ to every pair of actors $(x, y) \in V^2$.

Usually, C is the adjacency matrix of the network. Each of the entries $C(x, y)$ in C quantifies the strength of the relationship for the pair of actors x, y in a network V. The affinity between two actors shows how strongly they are connected according to different criteria, depending on which aspect of the relationship we are taking into account. A 0 affinity value means that no affinity has been found at all while an 1 value means that there is a perfect match according to the analyzed factors.

In the following, we recall the definition of two affinity functions (additional affinity functions can be found in [26] and [56]):

- Best friend affinity: it measures the importance of a relationship with an agent y for the agent x, in relation to all the other relationships of x:

$$F_C^{BF}(x, y) = \frac{C(x, y)}{\sum_{a \in V} C(x, a)}. \tag{20}$$

- Best common friend affinity: it measures the importance of the relationship taking into account how important are the common connections between the connected nodes to x and y, in relation to all other relationships of x in the network:

$$F_C^{BCF}(x, y) = \frac{\max_{a \in V}\{\min\{C(x, a), C(y, a)\}\}}{\sum_{a \in V} C(x, a)}. \tag{21}$$

For example, in the case of Best common friend affinity, 0 value means that there is no common connections between the two actors and a high value means that their shared friends are important to both of them. Since affinities are not necessarily symmetrical, the strength of this interaction depends on who the sender and receiver are, as it happens in human interactions e.g. unrequited love. So, it is possible that actor x has an affinity of value 1 with y, while y has a much lower value with x.

7.3. Interval-valued affinity functions

We define interval-valued affinity functions (IV-affinity functions) as functions that characterize the relationship between two actors, x, y with an interval in the $[0, 1]$ range, where the width of that interval represents the difference in commitment between two parties. We construct an interval-valued affinity function using a previously computed numerical affinity function. Then, the interval is constructed as:

$$F_{C,IV}(x, y) = [\min\{F_C(x, y), F_C(y, x)\}, \max\{F_C(x, y), F_C(y, x)\}] \tag{22}$$

Because $F_C(x, y) \neq F_C(y, x)$ in most relationships, this means that in most of them the actors give different levels of commitment than their counterparts. In real life, these kinds of situations are usually solved by finding a compromise between both parties. The IV-affinity function models this idea, representing the actual relationship that it is formed with an interval that ranges from both levels of commitment. The interval models the fact that we know that the final compromise achieved by both actors should be between both commitment levels, but we do not know exactly the final compromise.

One of the main differences between IV-affinity functions and their numerical counterpart is that they are symmetric. This can be convenient, as it allows to represent the relationship between two parties with one interval instead of two numerical values. This also opens the possibility of using some of the existing methods that require symmetric matrices in social network analysis [55,57], while retaining the desirable properties of affinity functions, i.e. zeros-sum game philosophy or local-only interactions taken into account [58].

7.4. Using interval-valued affinity functions to construct centrality measures

In this section we present four different centrality measures using IV-affinity functions. Usually, centrality measures ponder the importance of each actor in the network. However, our proposed centrality measures characterize the tendency of each actor to form relationships that have very different levels of commitment, and if the actor tends to show more or less commitment than the other party in each of its relationships.

The proposed centrality measures are:

1. **Asymmetry** is the tendency of the actor to form relationships with different levels of commitment.
2. **Altruism** is the tendency of the actor to form relationships in which its level of commitment is bigger than the other party.
3. **Egoism** is the tendency of the actor to form relationships in which its level of commitment is lesser than the other party.
4. **Generosity** is the difference between altruism and egoism. A positive generosity means that overall, the actor tends to give more commitment in its relationships than the other part. A negative generosity means that the actor tends to give less commitment than the other part in a relationship.

We have considered two possibilities to compute these metrics: using the width of each IV-affinity function and aggregate those values, or aggregating the IV-affinity values using an IV-aggregation and then use the width of the aggregated interval. We have opted for the latter, because in this way we are taking into account that some intervals have the same width, but are different.

We are specially interested in this property in the case where two actors have the same average width for their respective IV-affinity values. In these occasions, we consider that the one that presents more variety in their IV-affinities is more asymmetric than the other. For example, if an actor x has 3 different IV-affinities, and all of the are $(0, 0.3)$, and we have other actor y that has as IV-affinities $(0, 0.3)$, $(0.4, 0.7)$ and $(0.7, 1.0)$. We should assign the highest asymmetry value to y, because this actor presents this level of asymmetry in its relationships in all the $[0, 1]$ range.

Taking these considerations into account, we compute these centrality measures using the following expressions, with an IV-aggregation function I.

1. **Asymmetry**:
$$A(x) = w\left(I(F_{C,IV}(x, y_1), \dots, F_{C,IV}(x, y_m))\right), \tag{23}$$
where $\{y_1, \dots, y_m\} = V \setminus \{x\}$.
2. **Altruism**:
$$L(x) = w\left(I(F_{C,IV}(x, y_1), \dots, F_{C,IV}(x, y_k))\right), \tag{24}$$
where $\{y_1, \dots, y_k\} \subseteq V \setminus \{x\}$ is the set of all actors y_j fulfilling $F_C(y_j, x) \leq F_C(x, y_j)$ for each $j = 1, \dots, k$.
3. **Egoism**:
$$E(x) = w\left(I(F_{C,IV}(x, y_1), \dots, F_{C,IV}(x, y_l))\right), \tag{25}$$
where $\{y_1, \dots, y_l\} \subseteq V \setminus \{x\}$ is the set of all actors y_j fulfilling $F_C(x, y_j) \leq F_C(y_j, x)$ for each $j = 1, \dots, l$.
4. **Generosity**:
$$S(x) = L(x) - E(x) \tag{26}$$

For our experiment, we have used as I the proposed IV Sugeno-like FG-functional, with $G(X_1, \dots, X_n) = \min\{\mathbf{1}, \sum_{i=1}^{n} X_i\}$, $F(X, y) = X \cdot (1 - y)$ and the cardinality measure $m(A) = \frac{|A|}{n}$ as the fuzzy measure. Note that in the three respective cases of centrality measures defined above, the arity n of applied IV Sugeno-like FG-functional I is m, k, l, respectively.

7.5. Experimental results for a social network

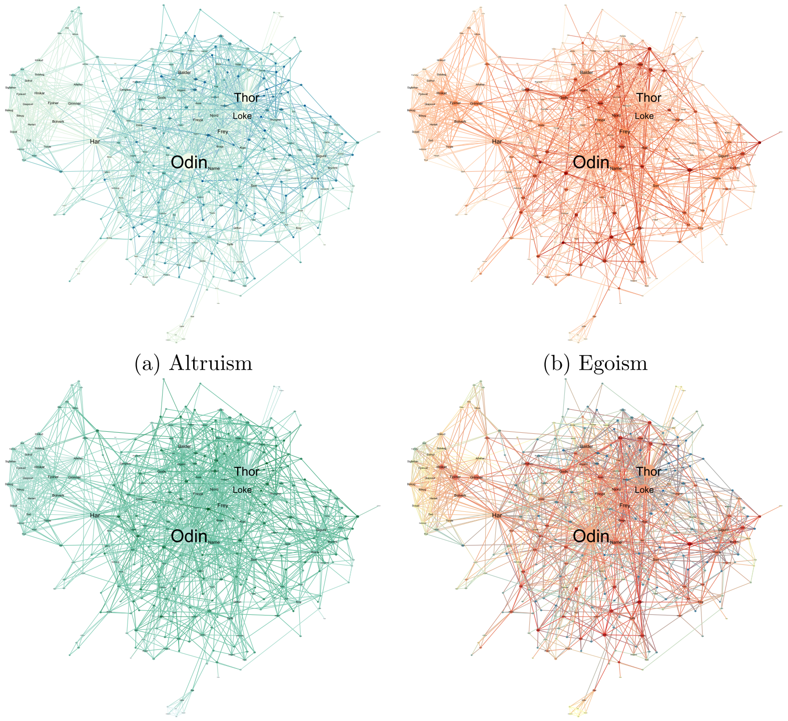
In this section we have studied the proposed centrality measures in a word association network, constructed using *The Younger Edda* book. *The Younger Edda* is Old Norse textbook of mythical texts, written approximately in 1220 by Snorri Sturluson. This book contains the tales of popular characters in the Nordic folklore, like Odin, Thor or Loki.

In order to extract the word association network from this text, we have followed the standard procedure to tokenize and lemmatize the text [59,60] and we have used a pre-trained multilayer perceptron in the Python Natural Language ToolKit [61] to purge every word that is not a noun, since we only want to model interaction between entities and concepts. Once we have extracted the nouns from the text, what we have is a series of stemmed tokens. To obtain a network, we need to determine its nodes and edges. In the case of the nodes, we make a bijective association, so that one noun corresponds to one node, and vice versa. There are different ways to compute the edges in terms of noun co-occurrence. We have decided to create an edge every time a word appears in a k-distance or less from another in the text, choosing k as 10. The text has been obtained from the Gutenberg Project [62].

We have computed the proposed centrality measures in this network: asymmetry, egoism, altruism and generosity. Fig. 2 shows the results in this network, coloring each node according the value of each metric. We found that the actors with highest degree tend to be less altruistic, like "Odin", "Thor", "Balder" or "Loki". However, low values in altruism do not necessarily imply high egoism values, as these same actors only showed moderate egoism values. The highest egoism values were located in actors like "gold" or "gods", that connected low degree actors with the ones with highest value. These actors also presented a moderate altruism value, that comes from their relationships with higher degree values. This means that the most egoist actors are willing to have a small number of connections where they are the "losing" part, because it allows them to have more relationships where they are the "favored" part.

Regarding generosity, most nodes have a negative value. The generous actors are low degree actors connected to higher degree actors, which in this case are concepts not very frequent in the original text. The less generous actors have small-medium degree values, generally connected to a small set of higher degree, high egoism actors, and a another set of very generous ones.

Table 5 shows the top values for each centrality measure. The highest asymmetry values correspond to actors like "Hammer", mostly connected with "Thor", a high degree actor; "Journey", connected to individual characters; and other general concepts. Altruism presents a similar top 10 of values, but with the presence of two individuals: "Sigurd", who is connected to one of the most egoist actors, "gold"; "Hermod", connected almost exclusively to characters with negative generosity like "Balder", "Odin" and "Æsir"; and "Hymir", who is connected to a similar set of actors to "Hermod". The most egoist actors are general concepts like 'Name", "Man", "Land" or "Gold". The only exception is "Atle" because he has only one relationship where he is the least committed part.



(a) Altruism (b) Egoism

(c) Asymmetry (d) Generosity

Fig. 2. Altruism (a), Egoism (b), Asymmetry (c) and Generosity (d) for the *Younger Edda* network, marked with different colors. Node size is proportional to the node degree. We can see that most actors are more egoist than altruist. In fact, some of the most important actors in this network show no altruism at all. However, no altruism does not necessarily imply a high egoism. For example, “Odin” and “Thor” show no altruism (so all their in-affinities are higher or equal than their out-affinities), but their asymmetry value is also low.

8. Conclusions and future lines

In this work we have proposed a new version of the generalized Sugeno integral to aggregate interval-valued data (IV-Sugeno). This function is designed to aggregate intervals taking into account the coalitions between the input data, just as the numerical Sugeno integral. We have also proposed two applications in which we use interval-valued data: a brain computer interface framework where the intervals measure the uncertainty related to the output of different classifiers; and social network analysis, where the intervals measure the difference in commitment in a relationship, and how this can be used to construct a centrality measure to characterize each actor.

Our results show that the generalized IV-Sugeno aggregation performs better than other IV-aggregations for a brain computer interface classification network. We have also found that the functions used to construct the generalized IV-Sugeno are critical in its performance and its mathematical properties. Finally, we have also shown how the generalized IV-Sugeno can be used to successfully characterize each actor in a network depending on the asymmetry in its relationships.

Future research shall study the use of the proposed IV-Sugeno in other settings, like image processing; and the study of other fuzzy integrals in the interval-valued setting. We also intend to study the possibilities of using IV-affinity functions with classical centrality measures in social network analysis and the possible correlations of the newly centrality measures proposed with classical ones.

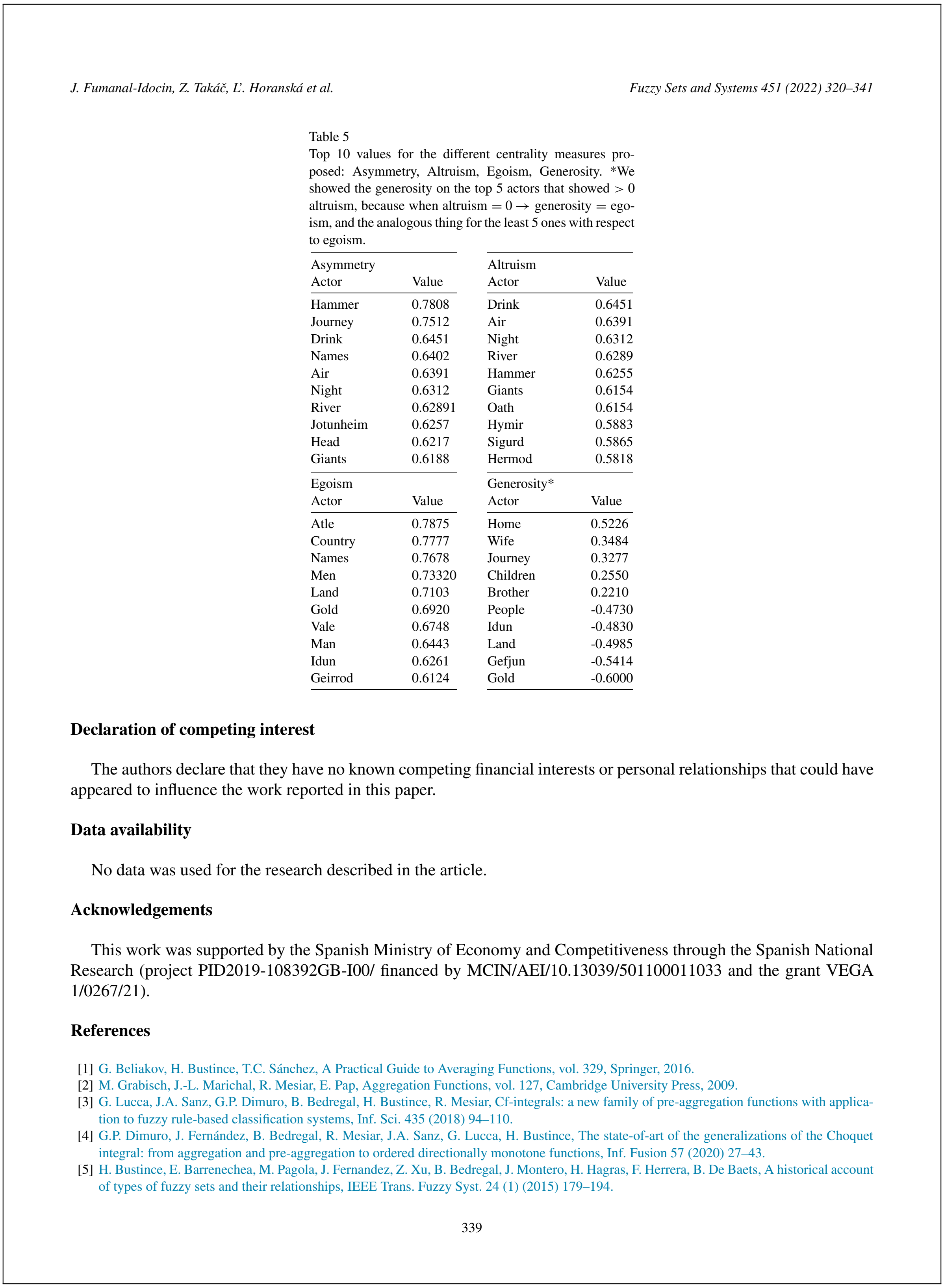


Table 5
Top 10 values for the different centrality measures proposed: Asymmetry, Altruism, Egoism, Generosity. *We showed the generosity on the top 5 actors that showed > 0 altruism, because when altruism $= 0 \rightarrow$ generosity $=$ egoism, and the analogous thing for the least 5 ones with respect to egoism.

Asymmetry		Altruism	
Actor	Value	Actor	Value
Hammer	0.7808	Drink	0.6451
Journey	0.7512	Air	0.6391
Drink	0.6451	Night	0.6312
Names	0.6402	River	0.6289
Air	0.6391	Hammer	0.6255
Night	0.6312	Giants	0.6154
River	0.62891	Oath	0.6154
Jotunheim	0.6257	Hymir	0.5883
Head	0.6217	Sigurd	0.5865
Giants	0.6188	Hermod	0.5818
Egoism		Generosity*	
Actor	Value	Actor	Value
Atle	0.7875	Home	0.5226
Country	0.7777	Wife	0.3484
Names	0.7678	Journey	0.3277
Men	0.73320	Children	0.2550
Land	0.7103	Brother	0.2210
Gold	0.6920	People	-0.4730
Vale	0.6748	Idun	-0.4830
Man	0.6443	Land	-0.4985
Idun	0.6261	Gefjun	-0.5414
Geirrod	0.6124	Gold	-0.6000

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was supported by the Spanish Ministry of Economy and Competitiveness through the Spanish National Research (project PID2019-108392GB-I00/ financed by MCIN/AEI/10.13039/501100011033 and the grant VEGA 1/0267/21).

References

[1] G. Beliakov, H. Bustince, T.C. Sánchez, A Practical Guide to Averaging Functions, vol. 329, Springer, 2016.
[2] M. Grabisch, J.-L. Marichal, R. Mesiar, E. Pap, Aggregation Functions, vol. 127, Cambridge University Press, 2009.
[3] G. Lucca, J.A. Sanz, G.P. Dimuro, B. Bedregal, H. Bustince, R. Mesiar, Cf-integrals: a new family of pre-aggregation functions with application to fuzzy rule-based classification systems, Inf. Sci. 435 (2018) 94–110.
[4] G.P. Dimuro, J. Fernández, B. Bedregal, R. Mesiar, J.A. Sanz, G. Lucca, H. Bustince, The state-of-art of the generalizations of the Choquet integral: from aggregation and pre-aggregation to ordered directionally monotone functions, Inf. Fusion 57 (2020) 27–43.
[5] H. Bustince, E. Barrenechea, M. Pagola, J. Fernandez, Z. Xu, B. Bedregal, J. Montero, H. Hagras, F. Herrera, B. De Baets, A historical account of types of fuzzy sets and their relationships, IEEE Trans. Fuzzy Syst. 24 (1) (2015) 179–194.

[6] I.J. Rudas, E. Pap, J. Fodor, Information aggregation in intelligent systems: an application oriented approach, Knowl.-Based Syst. 38 (2013) 3–13.

[7] G. Beliakov, H. Bustince, D. Paternain, Image reduction using means on discrete product lattices, IEEE Trans. Image Process. 21 (3) (2012) 1070–1083.

[8] J. Armentia, I. Rodríguez, J. Fumanal Idocin, H. Bustince, M. Minárová, A. Jurio, Gravitational clustering algorithm generalization by using an aggregation of masses in Newton law, in: R. Halaš, M. Gagolewski, R. Mesiar (Eds.), New Trends in Aggregation Theory, Springer International Publishing, Cham, 2019, pp. 172–182.

[9] R.R. Yager, Generalized OWA aggregation operators, Fuzzy Optim. Decis. Mak. 3 (1) (2004) 93–107.

[10] M. Grabisch, The application of fuzzy integrals in multicriteria decision making, Eur. J. Oper. Res. 89 (3) (1996) 445–456.

[11] J. Fumanal-Idocin, Z. Takac, J. Fernandez, J.A. Sanz, H. Goyena, C.-T. Lin, Y. Wang, H. Bustince, Interval-valued aggregation functions based on moderate deviations applied to motor-imagery-based brain computer interface, IEEE Trans. Fuzzy Syst. (2021).

[12] J. Fumanal-Idocin, Y.-K. Wang, C.-T. Lin, J. Fernández, J.A. Sanz, H. Bustince, Motor-imagery-based brain-computer interface using signal derivation and aggregation functions, IEEE Trans. Cybern. (2021).

[13] M. Grabisch, M. Roubens, Application of the Choquet integral in multicriteria decision making, in: Fuzzy Measures and Integrals - Theory and Applications, 2000, pp. 348–374.

[14] M. Grabisch, M. Sugeno, T. Murofushi, Fuzzy Measures and Integrals: Theory and Applications, Physica, Heidelberg, 2000, p. 2010.

[15] G. Lucca, J.A. Sanz, G.P. Dimuro, E.N. Borges, H. Santos, H. Bustince, Analyzing the performance of different fuzzy measures with generalizations of the Choquet integral in classification problems, in: 2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2019, pp. 1–6.

[16] R.R. Yager, J. Kacprzyk, The Ordered Weighted Averaging Operators: Theory and Applications, Springer Science & Business Media, 2012.

[17] B. Llamazares, Constructing Choquet integral-based operators that generalize weighted means and OWA operators, Inf. Fusion 23 (2015) 131–138.

[18] T. Murofushi, M. Sugeno, Fuzzy t-conorm integral with respect to fuzzy measures: generalization of Sugeno integral and Choquet integral, Fuzzy Sets Syst. 42 (1) (1991) 57–71.

[19] G.P. Dimuro, G. Lucca, B. Bedregal, R. Mesiar, J.A. Sanz, C.-T. Lin, H. Bustince, Generalized cf1f2-integrals: from Choquet-like aggregation to ordered directionally monotone functions, Fuzzy Sets Syst. 378 (2020) 44–67.

[20] S. Auephanwiriyakul, J.M. Keller, P.D. Gader, Generalized Choquet fuzzy integral fusion, Inf. Fusion 3 (1) (2002) 69–85.

[21] G.P. Dimuro, J. Fernández, B. Bedregal, R. Mesiar, J.A. Sanz, G. Lucca, H. Bustince, The state-of-art of the generalizations of the Choquet integral: from aggregation and pre-aggregation to ordered directionally monotone functions, Inf. Fusion 57 (2020) 27–43.

[22] M. Boczek, L. Jin, M. Kaluszka, Interval-valued seminormed fuzzy operators based on admissible orders, Inf. Sci. 574 (2021) 96–110.

[23] X. Pu, R. Mesiar, R.R. Yager, L. Jin, Interval Sugeno integral with preference, IEEE Trans. Fuzzy Syst. 28 (3) (2019) 597–601.

[24] M. Boczek, L. Jin, M. Kaluszka, The interval-valued Choquet-Sugeno-like operator as a tool for aggregation of interval-valued functions, Fuzzy Sets Syst. 448 (2022) 35–48.

[25] F. Bardozzo, B. de la Osa, L. Horanská, J. Fumanal-Idocin, M. delli Priscoli, L. Troiano, R. Tagliaferri, J. Fernandez, H. Bustince, Sugeno integral generalization applied to improve adaptive image binarization, Inf. Fusion 68 (2021) 37–45.

[26] J. Fumanal-Idocin, A. Alonso-Betanzos, O. Cordón, H. Bustince, M. Minárová, Community detection and social network analysis based on the Italian wars of the 15th century, Future Gener. Comput. Syst. 113 (2020) 25–40.

[27] S. Michio, Theory of fuzzy integrals and its applications, Doct. Thesis, Tokyo Institute of Technology, 1974.

[28] H. Bustince, J. Fernández, A. Kolesárová, R. Mesiar, Generation of linear orders for intervals by means of aggregation functions, Fuzzy Sets Syst. 220 (2013) 69–77.

[29] M.J. Asiain, H. Bustince, R. Mesiar, A. Kolesárová, Z. Takáč, Negations with respect to admissible orders in the interval-valued fuzzy set theory, IEEE Trans. Fuzzy Syst. 26 (2) (2017) 556–568.

[30] M. Akin, Comparison of wavelet transform and FFT methods in the analysis of EEG signals, J. Med. Syst. 26 (3) (2002) 241–247.

[31] T. Nierhaus, C. Vidaurre, C. Sannelli, K.-R. Mueller, A. Villringer, Immediate brain plasticity after one hour of brain–computer interface (BCI), J. Physiol. (2019).

[32] V.V. Nikulin, G. Nolte, G. Curio, A novel method for reliable and fast extraction of neuronal EEG/MEG oscillations on the basis of spatio-spectral decomposition, NeuroImage 55 (4) (2011) 1528–1535.

[33] S. Haufe, S. Dähne, V. Nikulin, Dimensionality reduction for the analysis of brain oscillations, NeuroImage 101 (2014) 583–597.

[34] C. Guger, H. Ramoser, G. Pfurtscheller, Real-time EEG analysis with subject-specific spatial patterns for a brain-computer interface (BCI), IEEE Trans. Rehabil. Eng. 8 (4) (2000) 447–456.

[35] A. Gramfort, M. Luessi, E. Larson, D.A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, et al., MEG and EEG data analysis with mne-python, Front. Neurosci. 7 (2013) 267.

[36] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, K.-R. Muller, Optimizing spatial filters for robust EEG single-trial analysis, IEEE Signal Process. Mag. 25 (1) (2008) 41–56.

[37] C. Sannelli, C. Vidaurre, K.-R. Müller, B. Blankertz, CSP patches: an ensemble of optimized spatial filters. An evaluation study, J. Neural Eng. 8 (2) (2011) 025012.

[38] C. Sannelli, C. Vidaurre, K.-R. Müller, B. Blankertz, Common spatial pattern patches - an optimized filter ensemble for adaptive brain-computer interfaces, in: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, 2010, pp. 4351–4354.

[39] K.-R. Müller, C.W. Anderson, G.E. Birch, Linear and non-linear methods for brain-computer interfaces, IEEE Trans. Neural Syst. Rehabil. Eng. 11 (2) (2003) 165–169.

[40] C. Vidaurre, R. Scherer, R. Cabeza, A. Schlögl, G. Pfurtscheller, Study of discriminant analysis applied to motor imagery bipolar data, Med. Biol. Eng. Comput. 45 (1) (2007) 61.

[41] A.J. Izenman, Linear discriminant analysis, in: Modern Multivariate Statistical Techniques, Springer, 2013, pp. 237–280.

[42] J. Fumanal-Idocin, Z. Takac, J. Fernandez, J.A. Sanz, H. Goyena, C.-T. Lin, Y. Wang, H. Bustince, Interval-valued aggregation functions based on moderate deviations applied to motor-imagery-based brain computer interface, IEEE Trans. Fuzzy Syst. (2021).

[43] C. Vidaurre, J. Pascual, A. Ramos-Murguialday, R. Lorenz, B. Blankertz, N. Birbaumer, K.-R. Müller, Neuromuscular electrical stimulation induced brain patterns to decode motor imagery, Clin. Neurophysiol. 124 (9) (2013) 1824–1834.

[44] C. Vidaurre, A.R. Murguialday, S. Haufe, M. Gómez, K.-R. Müller, V. Nikulin, Enhancing sensorimotor bci performance with assistive afferent activity: an online evaluation, NeuroImage 199 (2019) 375–386.

[45] C. Vidaurre, T. Jorajuría, A. Ramos-Murguialday, K.-R. Müller, M. Gómez, V. Nikulin, Improving motor imagery classification during induced afferent motor perturbations, J. Neural Eng. 18 (2021) 0460b1.

[46] C. Sannelli, C. Vidaurre, K. Müller, B. Blankertz, A large scale screening study with a SMR-based BCI: categorization of BCI users and differences in their SMR activity, PLoS ONE 14 (2019) e0207351.

[47] C. Vidaurre, S. Haufe, T. Jorajuría, K.-R. Müller, V.V. Nikulin, Sensorimotor functional connectivity: a neurophysiological factor related to BCI performance, Front. Neurosci. 14 (2020) 1278.

[48] M. Kawanabe, W. Samek, K.-R. Müller, C. Vidaurre, Robust common spatial filters with a maxmin approach, Neural Comput. 26 (2) (2014) 349–376.

[49] A. Chowdhury, J. Andreu-Perez, Clinical brain-computer interface challenge 2020 (CBCIC at WCCi2020): overview, methods and results, IEEE Trans. Med. Robotics Bionics (2021).

[50] V.J. Lawhern, A.J. Solon, N.R. Waytowich, S.M. Gordon, C.P. Hung, B.J. Lance, EEGnet: a compact convolutional neural network for EEG-based brain–computer interfaces, J. Neural Eng. 15 (5) (2018) 056013.

[51] S.R. Tibor, S.J. Tobias, F.L.D. Josef, G. Martin, E. Katharina, T. Michael, H. Frank, B. Wolfram, B. Tonio, Deep learning with convolutional neural networks for EEG decoding and visualization, Hum. Brain Mapp. 38 (11) (2017) 5391–5420.

[52] M. Hersche, T. Rellstab, P.D. Schiavone, L. Cavigelli, L. Benini, A. Rahimi, Fast and accurate multiclass inference for MI-BCIs using large multiscale temporal and spectral features, in: 2018 26th European Signal Processing Conference (EUSIPCO), 2018, pp. 1690–1694.

[53] M. Vijay, A. Kashyap, A. Nagarkatti, S. Mohanty, R. Mohan, N. Krupa, Extreme gradient boosting classification of motor imagery using common spatial patterns, in: 2020 IEEE 17th India Council International Conference (INDICON), 2020, pp. 1–5.

[54] A. Landherr, B. Friedl, J. Heidemann, A critical review of centrality measures in social networks, Bus. Inf. Syst. Eng. 2 (6) (2010) 371–385.

[55] M. Newman, Networks, Oxford University Press, 2018.

[56] J. Fumanal-Idocin, O. Cordón, G. Dimuro, M. Minárová, H. Bustince, The concept of semantic value in social network analysis: an application to comparative mythology, arXiv preprint, arXiv:2109.08023, 2021.

[57] X. Shi, H. Lu, Y. He, S. He, Community detection in social network with pairwisely constrained symmetric non-negative matrix factorization, in: Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015, 2015, pp. 541–546.

[58] Y.B. Reddy, Role of game models in social networks, in: 2009 International Conference on Computational Science and Engineering, vol. 4, IEEE, 2009, pp. 1131–1136.

[59] J.J. Webster, C. Kit, Tokenization as the initial phase in NLP, in: Proceedings of the 14th Conference on Computational Linguistics, vol. 4, COLING '92, Association for Computational Linguistics, 1992, pp. 1106–1110.

[60] C. Fox, A stop list for general text, ACM SIGIR Forum 24 (1–2) (1989) 19–21.

[61] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit, O'Reilly Media, Inc., 2009.

[62] B. Stroube, Literary freedom: project Gutenberg, XRDS: crossroads, ACM Mag. Stud. 10 (1) (2003) 3.

6.9 Quantifying External Information in Social Network Analysis: an Application to Comparative Mythology

Associated publication:

- Fumanal-Idocin, J., Cordón O., Dimuro G., Lopez-de-Hierro A.F. & Bustince, H. (2022).Quantifying External Information in Social Network Analysis: an Application to Comparative Mythology. *IEEE Transactions on Cybernetics.*

  Status: Published.

  Impact Factor (JCR 2021): 19.118.

  Categories:

  Computer Science, Artificial Intelligence. Ranking 3/144 (Q1).

  Automation & Control Systems. Ranking 1/65 (Q1).

  Computer Science, Cybernetics. Ranking 1/24 (Q1).

Fumanal-Idocin, J., Cordon, O., Dimuro, G. P., Lopez-de-Hierro, A.-F. R., & Bustince, H. (2023). Quantifying external information in social network analysis: An application to comparative mythology. *IEEE Transactions on Cybernetics*, 1-0. https://doi.org/10.1109/TCYB.2023.3239555 the final result taking into account the original difference in their semantic values S and the average affinity value in the edges used to carry it.

Since some actors naturally have low affinity values, for example when they have a lot of connections, the expected value of the average affinity values of a path can be deceptively low. In order to better compare the different semantic affinities that originate from actor x, we rescale the result by the maximum semantic affinity that x emits. This leads us to the final expression of $A(x, y)$:

$$A(x,y) = \left(1 - \frac{|S(x) - S(y)|}{\max(S(x), S(y))}\right) \frac{\sum P}{|P|} \cdot \frac{1}{\vee_{z \in Z} F_C(x,z)} \tag{10}$$

where P is the list of affinity values in the edges used in the paths to carry the semantic value, $S(x)$, from x to y, and Z is the set of all actors connected to x.

The pseudo-code for the pipe comparison algorithm is in Algorithm 1 and in Figure 1 we show a graphical example of one execution of this algorithm.

In order to compute the semantic affinity in our experimentation, we have used a combination of best friend and Machiavelli affinities as the $F_C(x, y)$. Using this mix of affinity functions we can characterize each edge based on the importance of the pairwise relationship between x and y, and also take into account the relative importance that both of their social circles have in the network. This is important for two reasons:

1) In high degree actors, the best friend affinity values are necessarily low, which will result in artificially low semantic values.
2) The Machiavelli affinity gives high affinity to actors that play a similar role in the network. In the texts we are studying, these results in high affinity values to concepts that play a similar role in the tales.

So, it is natural to think that those actors are transmitting information between them, even though the best friend affinity value between them is not high. We have combined both affinity functions using a convex combination, so the value of each edge is 90% the best friend affinity value and 10% the Machiavelli affinity, but we set to 0 all affinity values in the edges where the original Best friend affinity was 0.

V. Comparative mythology analysis using the semantic value and the semantic affinity

In this section we discuss the different steps to perform our mythology analysis:

1) How we constructed the network for each mythology, and how we fused them.
2) The results for the centrality measures and the semantic value in each network.
3) The affinity values for the semantic, best friend and Machiavelli affinities for important characters in their respective mythologies.

Algorithm 1 Pipe Comparison Algorithm

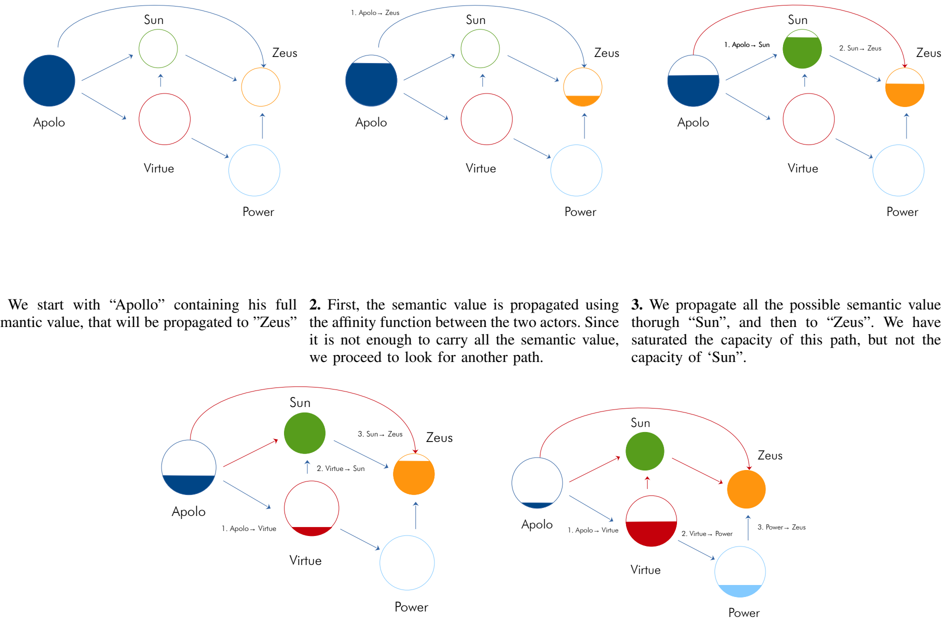
```
function FILLPATH($c$, $x$, $y$, $edgesSeen$, $F_C$, $M$)
    if $M(c(0)) \leq 0$ then
        return []
    end if
    $lenPath = length(c) - 1$
    $liquidCarried \leftarrow 0$
    $restingLiquid \leftarrow M(x)$
    for $i \in range(lenPath)$ do
        $afPath \leftarrow F_C(c(i), c(i+1))$
        $carry \leftarrow min(m(c(i+1), afPath * restingLiquid)$
        $M(c(i+1)) \leftarrow max(0, capacities(c(i+1)) - afPath * carry)$
        $liquidCarried \leftarrow liquidCarried + carry$
        $append(edgesSeen, ((c(i), c(i+1)))$
    end for
    $M(c(0)) \leftarrow M(c(0)) - liquidCarried$
    return $edgesSeen$
end function
function PIPECOMPARISON($G$, $x$ , $y$)
    $F_C \leftarrow buildAffinityNetwork(G)$
    $M \leftarrow computeSemantics(G)$
    $shortestPaths \leftarrow allEfficientPaths(F_C, x, y)$
    $edgesSeen \leftarrow []$
    for $c \in allEfficientPaths$ do
        $append(edgesSeen, fillPath(c, x, y, edgesSeen, F_C, M))$
    end for
    return $(1 - |M(x) - M(y)|/\max(M(x), M(y))) * average(edgesSeen)/\max(Af(x, :))$
end function
```

A. Building the mythology networks

In this section we show how we built the network for each mythology. We discuss which books were used to form each network, some statistics regarding word counts, and how we processed the text to obtain the desired networks.

1) Processing the texts: We have chosen three of the ancient mythologies to perform the comparative study: Greek, Nordic, and Celtic. We have opted for these three due to their well-known interest and the existence of available compilations of tales translated to English, which makes the text processing for each book much easier. We have chosen selected the following books as a basis for our analysis:

- *Celtic Wonder-Tales* by Ella Young (1867-1956) [64]. Originally written in 1910, it is a collection of Celtic traditional tales translated to modern English.
- *Greek Myths* by Olivia Collidge (1908-2006) [65]. Is a compilation of various stories regarding the classical Greek pantheon in modern English.
- *The Younger Edda* by Snorri Sturluson (1179-1241) [66]. *The Prose Edda* or *The Younger Edda* is a medieval Icelandic compilation of mythical texts, made by Snorri Sturluson, who was a historian, politician and poet in Iceland [67]. The degree of originality that he added to



1. We start with “Apollo” containing his full semantic value, that will be propagated to ”Zeus”

2. First, the semantic value is propagated using the affinity function between the two actors. Since it is not enough to carry all the semantic value, we proceed to look for another path.

3. We propagate all the possible semantic value thorugh “Sun”, and then to “Zeus”. We have saturated the capacity of this path, but not the capacity of ‘Sun”.

4. We use the affinity between “Apollo” and “Virtue”, and between “Virtue” and “Sun” to fill the rest of “Sun”. We have filled “Sun” completely, but we can still propagate thorugh “Virtue”.

5. We fill “Zeus” using the affinity between “Apollo” and “Virtue”, “Virtue” and “Power” and “Power” to “Zeus”. Some semantic value is still in Apollo, but “Zeus” is already filled so we cannot propagate more.

Fig. 1: **Example for a simplified execution of the Pipe algorithm to compute the semantic affinity between “Apollo” and “Zeus”**. Each subfigure describes the multiples steps taken to carry the semantic value from one actor to another. In this case we were able to fill “Zeus” completely, but we needed to use all the possible affinity values in the network, and some semantic value from “Apollo” did not fit in “Zeus”.

this compilation is unclear but the original stories contain material from traditional sources, reaching the Viking Age.

Since we have the plain text files, it is easy to extract each chapter/tale in each book. We then parse each of them following the standard procedure [68], [69] using a pre-trained multilayer perceptron in the Python Natural Language ToolKit [70]. We purge every word that is not a noun, since we only want to model interaction between entities and concepts. In Table I we report the size of each book and the number of entities found. All the texts have been obtained from the Gutenberg Project [71].

2) Obtaining the networks: Once we have extracted the nouns from the text, what we have is a series of stemmed tokens. To obtain a network, we need the nodes and the edges to form it. In the case of the nodes, we will make a bijective association, so that one noun will correspond to one node, and *vice versa*. There are different ways to compute the edges in terms of noun co-occurrence. We have decided to create an edge every time a word appears in a k-distance or less from another in the text, choosing k as 10.

TABLE I: **Report of the size of each mythology.** Number of words, chapters and entities for each book in this work.

Mythology	Book	Chapters	Words	Entities
Celt	*Celtic Wonder-Tales*	13	41613	5114
Greek	*Greek Myths*	27	61246	5985
Nordic	*The Younger Edda*	21	65388	7521

3) Fusing the networks: Given the network for each tale, we can fuse them to obtain a “global” network containing the information from all the different networks referring to each tale. Since many of these stories share a fair group of

TABLE II: **Entity intersection for the three cultures.** Percentage of entities in each culture that also are present in another one.

Common entities	Celt	Greek	Nordic
Celt	-	19.59%	17.20%
Greek	19.59%	-	23.08%
Nordic	17.20%	16.69%	-

TABLE III: **Centrality measures in *Greek Myths* network**. For the top 10 most repeated entities in the associated texts.

	S	E	Freq. (I)	Degree	Betweenness	Closeness	Eigencentrality
Heracles	239.49	89.49	150	364	0.48	0.52	0.53
Theseus	84.69	29.69	55	118	0.10	0.44	0.21
King	84.15	27.15	57	55	0.03	0.42	0.13
Jason	76.93	27.93	49	88	0.06	0.37	0.13
Apollo	64.83	20.83	44	86	0.08	0.40	0.13
Psyche	64.72	15.72	49	91	0.07	0.39	0.13
Eurystheus	57.54	25.54	32	77	0.01	0.39	0.16
Zeus	57.27	21.27	36	75	0.09	0.43	0.14
Perseus	45.30	16.30	29	54	0.04	0.36	0.08
Pelias	43.76	21.76	22	48	0.02	0.37	0.09

topics, they all have a significant amount of common terms with the others (Table II). There are no problems of scale in this context, since all stories range from 2 to 7 pages long only. So, we simply add up all the edges into a single network. When an edge between two actors is repeated in various of the networks, we take the highest value.

B. Analysis of the semantic value and centrality measures in the myth networks

We computed the word association networks for each of the mythologies studied and then we fused the networks into a single one, which are shown in Figure 2. Then, we computed the intrinsic, extrinsic and semantic value for each actor. We also computed other common centrality measures social network analysis [73]. We showed these results for the top most important actors according to semantic value in Tables III, IV, V and VI.

TABLE IV: **Centrality measures in *Celtic Wonder-Tales* network**. For the top 10 most repeated entities in the associated texts.

	S	E	Freq. (I)	Degree	Betweenness	Closeness	Eigencentrality
Lugh	130.73	55.73	75	258	0.13	0.51	0.27
Ireland	107.93	56.93	51	256	0.24	0.56	0.32
Conary	101.44	48.44	53	198	0.11	0.49	0.13
King	79.73	40.73	39	133	0.07	0.50	0.17
Son	78.22	41.22	37	176	0.06	0.49	0.18
Balor	73.10	38.10	35	186	0.08	0.49	0.20
Gobhaun	71.95	23.95	48	147	0.04	0.46	0.17
Ethaun	63.92	27.92	36	100	0.04	0.44	0.09
Fomor	56.07	24.07	32	100	0.03	0.47	0.17
Turann	54.17	22.17	32	105	0.03	0.44	0.11

TABLE V: **Centrality measures in *The Younger Edda* network.** For the top 10 most repeated entities in the associated texts.

	S	E	Freq. (I)	Degree	Betweenness	Closeness	Eigencentrality
Odin	215.00	106.00	109	1113	0.47	0.75	0.37
Thor	176.63	44.63	132	508	0.14	0.61	0.26
Loki	100.58	34.58	66	291	0.06	0.56	0.22
King	53.63	13.63	40	79	0.02	0.49	0.07
Frey	51.01	20.01	31	167	0.02	0.53	0.17
Har	50.24	16.24	34	116	0.03	0.52	0.09
Sigurd	45.02	19.02	26	178	0.02	0.52	0.13
Balder	43.69	14.69	29	151	0.02	0.52	0.14
Freyja	28.78	10.78	18	154	0.01	0.51	0.16
Norse	24.06	4.06	20	37	0.00	0.46	0.04

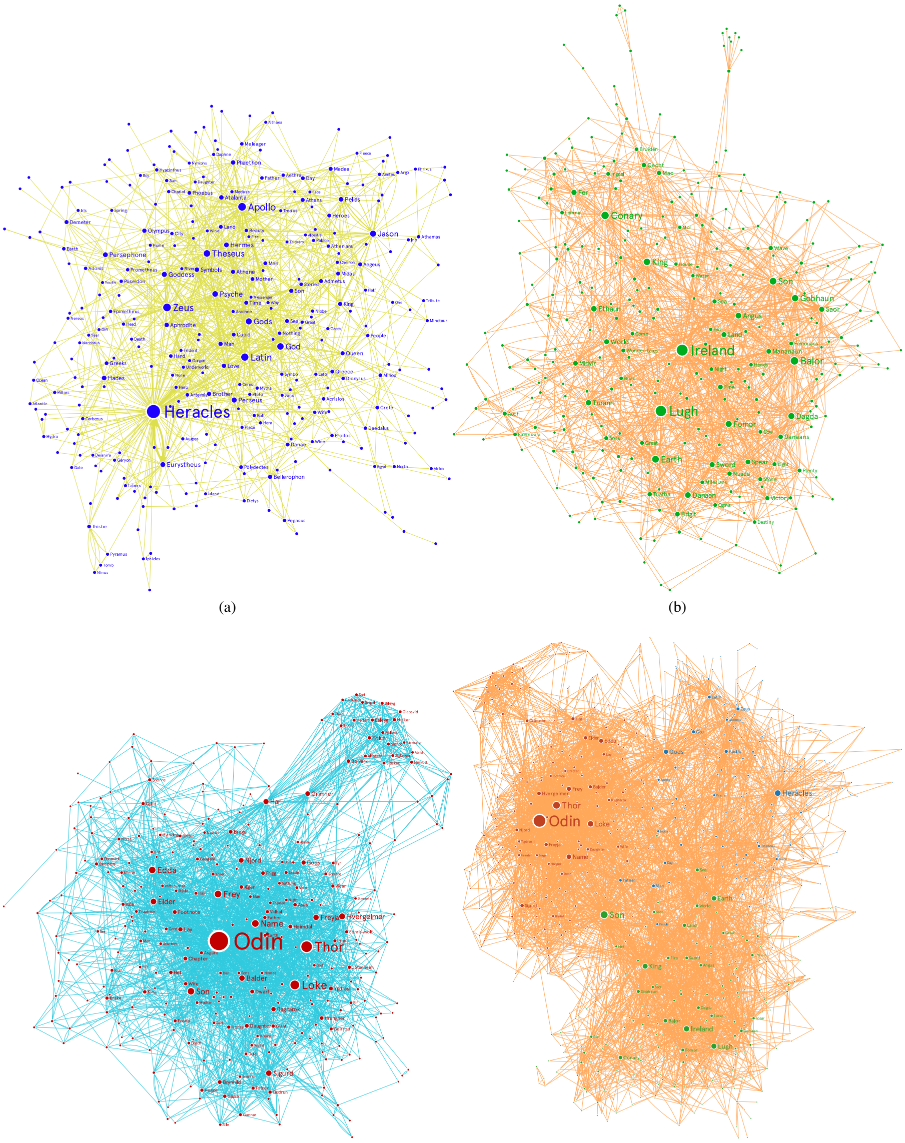
TABLE VI: **Centrality measures in the fusion network for the 10 most repeated entities in every text analyzed.**

	S	E	Freq. (I)	Degree	Betweenness	Closeness	Eigencentrality
Heracles	244.33	94.33	150	368	0.15	0.43	0.07
Odin	227.31	118.31	109	1121	0.17	0.48	0.33
King	222.12	86.12	136	269	0.09	0.48	0.13
Thor	178.63	46.63	132	502	0.05	0.44	0.24
Lugh	131.88	56.88	75	259	0.04	0.41	0.08
Son	116.97	77.97	39	414	0.11	0.49	0.22
Ireland	116.43	65.43	51	261	0.08	0.43	0.10
Conary	104.54	51.54	53	196	0.03	0.40	0.05
Loki	103.06	37.06	66	293	0.02	0.43	0.20
Theseus	80.19	25.19	55	116	0.02	0.40	0.04

We have found that “Heracles” is the most important actor in the *Greek Myths* network, according to all the measures taken. There are other important heroes in this list like “Theseus”, “Jason”, and “Perseus”. All of them are somewhat the embodiment of bravery and authority, so it is not surprising that “King” has also a high semantic value. There are more human characters than gods: “Apollo” and “Zeus” are the only ones which appear at the top, with similar S values, but not as high as the other Greek heroes here present. Regarding the classical centrality measures studied, the betweenness gives the highest value to “Hercules” by a large margin and penalizes specially “King” compared to the other metrics. The closeness does not show such a big gap between “Hercules” and the other actors, and similarly to the betweenness prefers “Zeus” over the human actors that posses more semantic value than him. This also happens in a smaller scale in the eigencentrality, that also preferred “Eurystheus” over the rest of the heroes. In general terms, classic centrality measures preferred gods, while the semantic value valued human and heroic figures (Table III).

In the case of *Celtic Wonder-Tales*, we found “Lugh”, the most prominent god of the Irish pantheon, to own the highest S value followed by “Ireland”. The third actor in S value, “Conary”, is an important mythical king of Ireland whose reign ends when he breaks three sacred oaths. The concept of “King” also has a high S, just as in the *Greek tales* case. Regarding the classical centrality measures, all of them rated most highly “Ireland”. Betweenness and closeness seem to be quite correlated in this case, showing the same top 3, but the betweenness quickly decreases after that. The eigencentrality values “Balor” significantly compared to the other metrics computed as it ranks in the third position. Contrary to most classical centrality measures, the semantic value favored the mythical embodiments of kingship like “Lugh”, “Conary”, “Balor”, and “Ethaun”. This fact, alongside the high S value of “Ireland”, indicates a strong connection in this compilation between these mythical figures and the sovereign of the country (Table IV). Such bond was not found in the other two mythologies.

In *The Younger Edda* we found “Odin”, one of the main gods of the Germanic pantheon, to be the most important actor in terms of S. Being the father of all the Æsir, but also wise in the ways of magic and divination, the strength of these two different attributions might be the origin of such high S



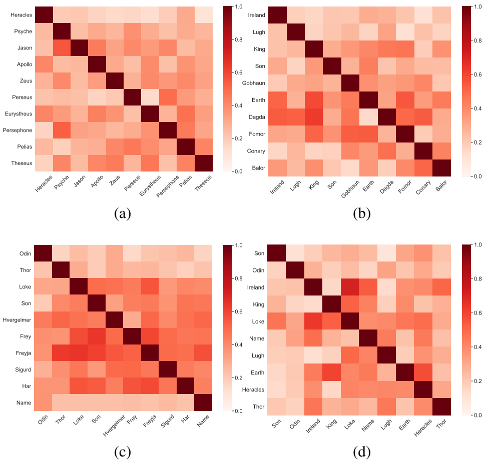
(a)

(b)

(c)

(d)

Fig. 2: **Word co-occurrence networks**. Each network is formed using the 300 most repeated entities in each corpus. We consider a connection between two words every time they appear less than 10 words apart from each other in one of the texts analyzed. **a.** *Greek Myths* **b.** *Celtic Wonder-Tales* **c.** *Younger Edda* **d.** Fusion network of the three cultures. For the fusion network, color is attributed to each node according to the frequency in each corpus. Red means majority of appearances in *Younger Edda*, blue in *Greek tales* and green in *Celtic Wonder-Tales*. Node size is directly proportional to the in-degree measure and the layout algorithm considered is Force Atlas 2 [72].



(a) (b) (c) (d)

Fig. 3: **Semantic affinities in all the networks studied**. We chose the 10 most repeated entities in each text to compare themselves. **a** *Greek myths* network. **b** *Celtic Wonder-Tales* network. **c** *Younger Edda* network. **d** Fused myths network.

value. Following "Odin", there is "Thor", another character with many attributions in his tales. A total of 6 gods populate this ranking, which shows that the gods themselves are more important in this mythology than in the other two. Regarding the four classical centrality measures, the top 3 is the same than in the semantic value. After the top 3, all the metrics prefer characters rather than concepts, but the semantic value puts "king" as the top 4 value, in consonance with the other compilations, which is an important difference (Table V).

Finally, for the case of the fused network, we can see how the three fused structures can be recognized in the final structure but also numerous bridges have appeared to join them. As expected, these bridges are mostly general concepts, such as "Son", "Gods" and "Father", which connect the specific deities for each mythology. When analyzing the semantic values, "Odin" is again the one with the highest value. However, in this case the correlation of I and S seems to be less important. For example, "Son", with an $S = 78.12$, did not enter in the top 10 most frequent words, neither "Earth" with $S = 44.63$. Comparing the semantic value to the remaining classic centrality measures, "Odin" generally gets the top value for them, which again reinforces the tendency of the the semantic value to prefer human and heroic figures over deities. The eigencentrality and the degree centrality generally favored actors from the *Younger Edda*, probably because they form a more densely connected structure. The closeness put "Son" as the top value, since 'Son" is an important bridge between the Nordic and Celtic entities.

C. *Semantic affinity analysis in the myth networks*

In Fig. 3 we show how the 10 most repeated entities in each text, ordered according to semantic value, relate to each other in terms of semantic affinity and showcase some of the most interesting actors to study. There are some relationships to remark:

- "Psyche", the impersonation of the human soul and lover of "Eros", receives significant semantic affinity from "Persephone", who is the wife of "Hades", and "Jason". There is a direct connection between "Persephone" and "Psyche", as they both appear in the same story, and both are the wife of a god and both are connected to the underworld. However, there is not straightforward connection with "Jason".
- "Apollo" sends the most of his affinity to "Heracles", who is also a character related to many different virtues, and receives the most from "Jason". "Jason" is also an important hero, so it is natural that he is connected to the god of virtue. Besides, "Jason"'s mother was a lover of "Apollo", which might imply deeper connections between these two characters.
- "Zeus" is connected more strongly to human characters than other gods and its most important connection is with "Psyche", the personification of the human soul.

In Fig. 3(b) we show how the top 10 entities according to S value in the Celtic mythology network relate to each other in terms of semantic affinity. Some of the findings in this figure are:

- "Dagda", the sun god, emits most semantic affinity to "King"and "Ireland", which suggests a relationship between earthly and divine mandates. Besides, "Dagda" is heavily entwined with "Ireland" but not with "Earth", which implies a negative connotation for "Earth". This might be in line with the idea that good things are "heavenly" things and "bad" things are more "earthly".
- "Earth" emits a lot of semantic value to "King", reinforcing again the bond between the earth and the ruler.
- "Balor", the king of the Fomorians, emits most of his affinity to "Conary Mór", which is a prototype for a good king. However, this king also breaks three sacred oaths in his story and this might connect him with negative characters such as "Balor". "Balor" also emits significant semantic value to its tribe, "Fomor", which is expected, but also to "Earth" and "Ireland".

In Fig. 3(c) we show how the top 10 entities in S value in the Nordic mythology network relate to each other in terms of semantic affinity and remark some of the most interesting actors to study. We can observe that:

- Both "Odin" and "Thor" have very high semantic affinity values compared to other actors. This is probably due to the fact that these gods have many attributions. In the case of "Name", which does not have neither high semantic affinity values, it does have a higher semantic affinity to "Odin", probably for the high number of different names that this god has in the texts.
- "Freyja" emits and receives significant affinity from "Loki". "Loki" is the responsible for the death of the

almost invincible god “Baldr”, who is also “Freyja”’s son. “Freyja” is considered the leader of the Valkyries and takes half of the fallen to her own afterlife field. This high affinity value here might indicate that the death theme is in fact a very important bond between them.

- “Frey”, one of the most important Vanir gods, sends the most affinity to the actor “Son”. “Frey” is usually associated with sacral kingship and his name derived phonetically from old Norse means “Lord”. This probably says that the original sacramental attributions of this Vanir god were abandoned in favor of the attributes of the more popular Æsir.
- “Hvergelmer” is the fountain in Nifelheim, the reign of the dead from which all rivers are born, and it is mostly associated with chaos. “Hvergelmer” sends and receives a significant amount of affinity from “Freyja” and “Loki”, and both of them showed certain relationship with death. Besides, just as in the case of “Heracles” and “Hydra”, the most emitted semantic affinity is to “Thor”, which is considered to be associated to order.

In Fig. 3(d) we did the analogous experiments for the fusion network of the three mythologies. Among other possible interesting relationships, it is notable that:

- The highest affinity of “Lugh” is “Loki”. This is a remarkable result, as there have been many studies discussing a possible a relationship between these two gods [74].
- A strong semantic affinity between “Ireland” and “Loki”, in both directions, and between “Ireland” and “Thor”, to a lesser extent. This might be due to “Ireland” being notably close to “Lugh”, who is a god closely entwined to both “Odin” and “Loki”, and because all of them are symbols related to authority in their original stories.
- “Earth” is notably affine to “King”, which means that the strong tie between the land and the ruler present in the *Celtic Wonder-Tales* network is also present in the other two.

D. Semantic affinity compared to other affinities in relevant actors

To complete our analysis, we have focused on the affinity study of a series of important characters in the original material. We have decided to study three characters per mythology, each one of them chosen according to their popularity and cultural significance.

In Figure 4 we report the results for the best friend, Machiavelli, and semantic affinities for “Zeus”, “Athene”, and “Heracles”.

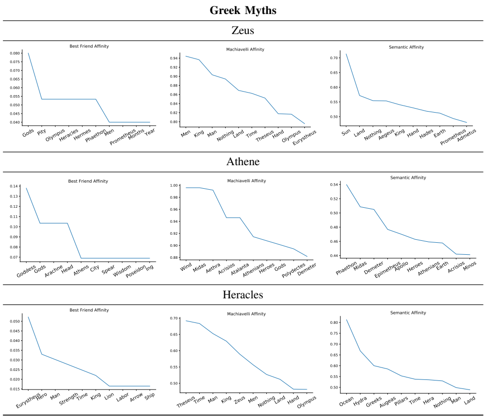


Fig. 4: **Study of affinities for three key characters in *Greek myths*.** Top 10 affinity values for the best friend, Machiavelli, and semantic affinity for “Zeus”, “Athene”, and “Heracles” in the *Greek Myths* network.

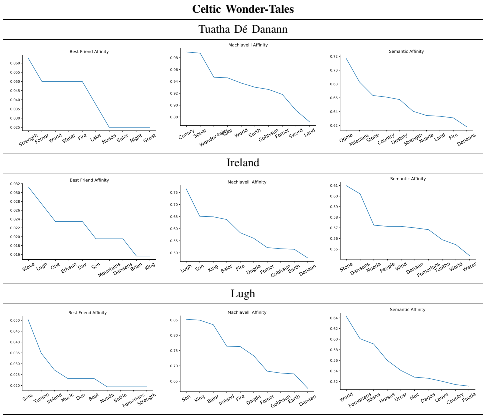


Fig. 5: **Study of affinities for three key characters in *Celtic Wonder-Tales*.** Top 10 affinity values for the best friend, Machiavelli, and semantic affinity for “Tuatha Dé Danann”, “Ireland”, and “Lugh” in the *Celtic Wonder-Tales* network.

In the case of “Zeus”, we can see that the best friend affinity includes mostly other Olympic gods. However, it is interesting to note that the results for the Machiavelli and semantic affinities do not show the same gods. This means that although “Zeus” appears repeatedly with other gods in his stories, he plays a very different role in the big picture. His highest semantic values reveal that he is mostly affine with general concepts, such as “Sun”, “Land”, “Nothing”, and “Time”, which indicates a connection between the world state and this god. The high affinities to “Sun” and “King” reinstate the connection of this god with the idea of authority.

For the case of “Athene”, her best friend affinity mostly connects her to characters that share her stories. However, the same thing happens as with “Zeus”. In the case of the semantic affinity, “Phaeton”, son of the sun-god “Helios”, is her highest affinity value, with no obvious connection between them. Following “Midas”, “Demeter”, is the goddess of agriculture and the afterlife, while “Athene” is also connected to agriculture through its association with olives and olive oil. Her top 4

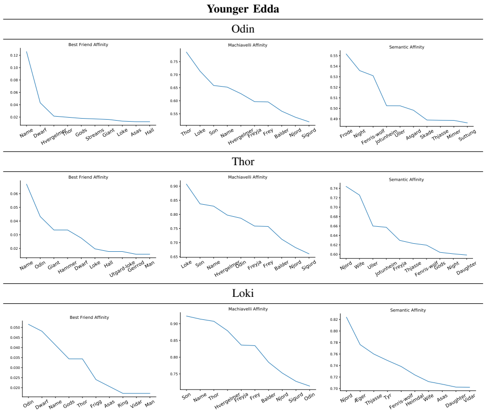


Fig. 6: **Study of affinities for three key characters in *The Younger Edda*.** Top 10 affinity values for the best friend, Machiavelli, and semantic affinity for “Odin”, “Thor”, and “Loki” in *The Younger Edda* network.

semantic affinity value, “Epimethius”, is the titan that has not foresight and he is only capable of pondering on what has already happened, which seems to be similar to wisdom, the most important attribute of “Athene”.

“Heracles” is the son of “Zeus” and “Alcmena”, a mortal queen, and he is one of the most important heroes in the Greek mythology. Many gods appear in his top best friend affinities, as his stories are filled with Olympic deities. His top Machiavelli affinities include “Weight” and “Labors”, which shows that the hero himself is tightly identified with his own narrative and his main characteristic: strength. The top 1 semantic affinity is “Ocean”. “Ocean” is a titan that guarded the limitless-like waters that surrounded the known world. The connection between “Heracles” and the differentiation between known and uncharted waters is present in modern-day symbology because of the Pillars of Hercules, a name that the Greeks stated for the geographic formations that surrounded the strait of Gibraltar. The top 2 semantic value, the “Hydra”, is the embodiment of chaos while the “Heracles” is the epitome of the Olympic order against it. This value shows us that these two characters are both impersonations of antagonistic ideas, so their connection is very strong.

We have also studied in Figure 5 the best friend, Machiavelli, and semantic affinity for three important actors in *Celtic Wonder-tales*: “Tuatha Dé Danann”, “Ireland”, and “Lugh”.

The “Tuatha Dé Danann” is the pantheon of pre-Christian gods in Gaelic Ireland that populate the stories in *Celtic Wonder-Tales*. Its top best friend affinity value is “Strength”, which reveals what is their most important attribution in these tales. The second value in best friend affinity is “Fomor” which is short for Fomorians. The Fomorians are the rival tribe of gods that represent the destructive face of nature, so the high affinity value reveals the importance of the conflict between the two tribes of gods in the text.

The “Cauldron” is one of the four artifacts linked to the “Tuatha Dé Danann”, alongside “Stone”, which also appears in the top 10 of best friend values. For the Machiavelli affinity, we can see that the top value is “Conary”, the mythical king of the Irish, seconded by “Dagda”. In the case of the semantic affinity, the greatest value is “Ogma”. “Ogma” is the god brother of “Dagda” and it is usually associated with him and with “Lugh”. He is considered to be the inventor of Ogham, an ancient alphabet. This remarkable attribution, alongside the relationships with these other two gods and with his strength, makes him a very versatile god, which maybe explains why the semantic value is so high. The “Stone” actor is referring to the Stone of Destiny, that is the stone in which the Kings of Ireland were crowned, which connects this tribe of gods with the sovereign of the land.

“Ireland” is one of the most repeated concepts in these tales, in contrast to the other two mythologies, where the origin country is not so clearly stated. “Ireland” is being put in these tales as one of the most important topics, always associated with figures of political relevance in the mythical landscape of these tales. If we look at the best friend and Machiavelli affinities, in both cases “Lugh” is one of the highest affinity values. The two most important semantic affinity values are “Stone” and “Danaans”, which are clearly referring to key actors in the designation of the ruler in that land as well. The third value, “Nuada”, is the first king of the “Tuatha Dé Danann”, which reinforces even more this idea.

Finally, “Lugh” is one of the most important gods in Irish mythology and also a member of the “Tuatha Dé Danann”. He is the maternal grandson of “Balor”, the leader of the Fomorians, which makes him a descendant of both tribes of gods in this mythology. His best friend affinity values show that he is indeed tightly connected to the “Tuatha Dé Danann” and other authoritarian symbols such as “Ireland”. His Machiavelli affinities show that indeed the structure of actors formed around him is similar to actors that wield authority, such as “King” and “Balor”. The semantic affinity reveals that the top value is “World”, which reveals how wide the attributions and roles for this god are. “Ildana”, “Lauve” and “Fauda” are other words to refer to “Lugh”.

In the Nordic mythology analysis, we have opted to showcase the results for the three most popular characters of these texts: “Odin”, “Loki”, and “Thor”. Their top 10 best friend, Machiavelli, and semantic affinities are collected in Figure 6.

“Odin”’s most important best friend affinities are “Name”, as he is introduced many times in *The Younger Edda* with different titles, and “Dwarf”, as the tribes of dwarves also appear repeatedly in the presence of “Odin” and share many attributions. The two most important Machiavelli affinities are his son “Thor” and “Loki”. The relationship of “Odin” and “Loki” is very complex and they both take a central role in the many stories of the corpus. We also found that his highest semantic affinity is “Frode”. “Frode” is another name for “Frey”, the leader of the other tribe of gods in *The Younger Edda*, the Vanir. “Frode” is associated with authority and sacral kingship, which explains why these gods are so affine in this case. We also found that “Night” has a very high semantic affinity with “Odin”. This is indeed quite an abstract connection but it is true that “Odin” is highly associated with the Wild Hunt, a repeated folklorical motif in which a group of supernatural hunters lead by a mythical figure chase the skies in the night [75]. He is also considered to be a god of the dead, as he greets fallen warriors in the Nordic afterlife, and he is also capable to raise dead out of the earth. He is also affine to “Fenris the wolf” and “Jotunheim”, which are enemies to “Odin” and key characters in the developing of the Ragnarök.

In the case of “Thor”, he has a high best friend affinity for “Name”, similarly to “Odin”, and “Odin” and “Loki” also appear in the top 10, as they are recurrent characters in his stories. For the Machiavelli affinity, “Loki” is again the most affine actor. In the semantic affinity, the top value is again a Vanir god, “Njord”, god of navigation and fertility. The second value is “Wife”. In one of *The Younger Edda* tales, “Thor” dresses as a bride, instead of “Freyja”, to try to recover his famous hammer, the “Mjolnir”. This tale could be the origin of this relationship, as the semantic affinity to “Freyja” is also high. “Ullr” is the stepson of “Thor”, associated with archery and skillfulness among other attributes, and there are evidences that he was a very important god in times before the cult of the Æsir. There are many similarities in both of these gods attributions, which is probably the reason for this high affinity value.

For “Loki”, the top best friend affinity is “Odin”, followed by “Dwarf”, because they appear repeatdly together in a set of stories of *The Younger Edda*. The Machiavelli affinity values are very similar to those of “Thor”, which emphasizes the fact that these two gods play a very similar role in the texts. The top semantic value of “Loki” is again “Njord”, followed by “Ægir”, who is the personification of the ocean and member of the Jötun tribe, just as “Thjasse”. “Tyr” is a member of the Æsir and the god of war, whose hand is bitten by “Fenris the wolf”, which is the next semantic value, and is also a son of “Loki”. The presence of both Jötunn and Æsir in the top 10 values without a clear identification with one of the tribes is a clear sign of the characteristic ambivalence of this character.

VI. Discussion and Conclusions

In this work we have studied the relationships between the meaning of different characters and concepts in three classical mythologies by extending Social Network Analysis to work with the concept of “meaning” or “Noumenon” [2]. We have done so by combining different affinity functions and some ideas from fuzzy logic, in order to characterize the semantic value in each actor. We have also proposed a new heuristic search algorithm, the Pipe algorithm, to compute the affinity between the semantic affinity of a pair of actors, that we called the semantic affinity. This algorithm uses a combination of affinity functions and a heuristic search in the network to compute the semantic affinity in an efficient way.

Using our proposal, we intend to model the way in which actors, words in our experimentation, enrich their own meanings by connecting with others; we can also compare the meanings of different actors or we can use them as adjectives to describe another actor. We compute a network for each culture studied, and another one obtained by fusing the three individual ones. We show the words with a more relevant meaning and the most important comparisons and similarities found, recovering already-known parallelism and similarities between them, and gaining new insights in the meaning and connotations of some of the tales studied.

Results in the three networks showed a mix of both historical and psychological relationships among the different actors in the network. Generally speaking, we found that gods are very close to kingship and authoritarian concepts, and particularly in the case of the Celtic myths, also to the land. We also found that gods serve as a common nexus between the different topics that appeared in each of the compilation tales, and the Nordics and Celts prefer to center their stories in gods, while the Greeks did so on humanoid and heroic figures.

When fusing all the tales together, we also observed that in the final networks three different structures, corresponding to the original tales, could be easily identified. Each one comprised mainly of the singular characters that appeared in only in of the compilations. The two most important actors that played a role of “bridge” between different mythologies were “Son” and “Gods”. Other important actors in that sense were “Man”, “Father” “Day”. Only “Son” appeared in the top 10 of semantic values. Contrary to our initial hypothesis, it seems that individual characters have more semantic value than general concepts. This might be explained by the high number of attributions that many of these gods and heroes have. Logically, it is easier to tell stories about humans or human-like beings rather than essays about trades or arts, which explains why these stories fill their characters with such many different traits.

When comparing semantic affinities, we found a strong bond between kingship and the earth. This has been previously studied in many traditions thorough the world [23] and it seems that it left its footprint in these tales as well. We also found the traces of historical connections between the Nordic gods “Loki” and “Odin” and the Celtic myths. This has been hypothesized before in [74], and the connection is clear in this analysis.

The main limit of this study has been the great amount of culture and relevant information that is not present in texts, but was important in the cult of each religion. This is not a problem for our new techniques, as they can work with information from outside the texts, as long as the initial network is constructed. Indeed, it is possible to construct affinity relationships using non-textual data, but it requires choosing the correct information to study and ponder carefully each source. Of course, adding more tales from different cultures is also a promising direction, but we must take into account dates, styles and original languages when doing so, or else an important deal of the findings could be more related to the translation than to the original material.

Semantic value can also be used to explore social interactions in other domains where a significant part of the information is outside the network itself, such as economic or trading networks. It also possible to use the semantic value to understand the interactions between actors in complex systems or how knowledge is propagated in networks of information [76], [77].

Code and Data Availability

Further results and the code to replicate our experiments can be found in this public repository: https://github.com/Fuminides/noumenon_project.

The original texts are freely available in Project Gutenberg [71].

References

[1] R. DiSalle, *Understanding space-time: The philosophical development of physics from Newton to Einstein*. Cambridge University Press, 2006.

[2] M. Friedman, *Kant's construction of nature: a reading of the metaphysical foundations of natural science*. Cambridge University Press, 2013.

[3] M. Heidegger, *Introduction to metaphysics*. Yale University Press, 2014.

[4] L. Wittgenstein, *Tractatus logico-philosophicus*. Routledge, 2013.

[5] ——, *Philosophical investigations*. John Wiley & Sons, 2010.

[6] G. Piccinini, *Computation in Physical Systems*, summer 2017 ed. Metaphysics Research Lab, Stanford University, 2017.

[7] B. F. L. Pozuelo, *The boundaries of machine intelligence*. University of Fribourg, 1969.

[8] P. E. Oppenheimer and E. N. Zalta, "A computationally-discovered simplification of the ontological argument," *Australasian Journal of Philosophy*, vol. 89, no. 2, pp. 333–349, 2011.

[9] L. A. Zadeh, "Fuzzy logic=computing with words," in *Computing with Words in Information/Intelligent Systems 1*. Springer, 1999, pp. 3–23.

[10] ——, *Computing with words in Information/Intelligent systems 1: Foundations*. Physica, 2013, vol. 33.

[11] ——, "From computing with numbers to computing with words. From manipulation of measurements to manipulation of perceptions," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 46, no. 1, pp. 105–119, 1999.

[12] E. I. Papageorgiou, "Learning algorithms for fuzzy cognitive maps—a review study," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 2, pp. 150–163, 2011.

[13] J. P. Carvalho, "On the semantics and the use of fuzzy cognitive maps and dynamic cognitive maps in social sciences," *Fuzzy Sets and Systems*, vol. 214, pp. 6–19, 2013.

[14] J. T. Rickard, J. Aisbett, D. G. Morgenthaler, and R. R. Yager, "Modeling of complex system phenomena via computing with words in fuzzy cognitive maps," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 12, pp. 3122–3132, 2019.

[15] Y. Wang, "On formal and cognitive semantics for semantic computing," *International Journal of Semantic Computing*, vol. 4, no. 2, pp. 203–237, 2010.

[16] P. Sheu, H. Yu, C. Ramamoorthy, A. K. Joshi, and L. A. Zadeh, "Semantic computing," 2010.

[17] N. Guarino, D. Oberle, and S. Staab, "What is an ontology?" in *Handbook on Ontologies*. Springer, 2009, pp. 1–17.

[18] L. Cameron and D. Larsen-Freeman, "Complex systems and applied linguistics," *International Journal of Applied Linguistics*, vol. 17, no. 2, pp. 226–239, 2007.

[19] P.-O. Siebers and U. Aickelin, "Introduction to multi-agent simulation," in *Encyclopedia of decision making and decision support technologies*. IGI Global, 2008, pp. 554–564.

[20] M. E. Stith and M. Leach, *The Beginning: Creation Myths around the World*. New York: Funk & Wagnalls, 1958.

[21] R. Reynolds, *Super heroes: A modern mythology*. Univ. Press of Mississippi, 1994.

[22] J. Campbell, *The hero with a thousand faces*. New World Library, 2008, vol. 17.

[23] J. G. Frazer, *The Golden Bough*. Palgrave Macmillan UK, 1922.

[24] W. H. Capps, *Religious studies: The making of a discipline*. Fortress Press, 1995.

[25] J. E. Cirlot, *Diccionario de símbolos (in Spanish)*. Siruela, 2004.

[26] S. S. Elliott, *Reinventing Religious Studies: Key Writings in the History of a Discipline*. Routledge, 2014.

[27] T.-S. B. Liew, *Present and Future of Biblical Studies: Celebrating Twenty-five Years of Brill's Biblical Interpretation*. Brill, 2018.

[28] A. Lassen, "Old norse mythology—comparative perspectives," *Journal of English and Germanic Philology*, vol. 118, no. 2, pp. 298–300, 2019.

[29] J. Bengtson, "Iarl and Iormun-; Arya-and Aryaman-: A study in indo-european comparative mythology," *Comparative Mythology*, vol. 2, no. 1, pp. 33–67, 2016.

[30] Y. Wen, "Norse mythology and chinese mythology in comparison: The origin of the cosmos, time and space," *Germanic Myths in the Audiovisual Culture*, vol. 5, p. 183, 2020.

[31] T. Keightley, *The Mythology of Ancient Greece and Italy*. G. Bell and Sons, 1877.

[32] W. Burkert, *Oriental and Greek mythology: the meeting of parallels*. Verlag nicht ermittelbar, 1987.

[33] B. G. Purzycki, "The minds of gods: A comparative study of supernatural agency," *Cognition*, vol. 129, no. 1, pp. 163–179, 2013.

[34] C. S. Littleton, *The new comparative mythology: an anthropological assessment of the theories of Georges Dumézil*. Univ of California Press, 1973.

[35] J. Dow, "Is religion an evolutionary adaptation?" *Journal of Artificial Societies and Social Simulation*, vol. 11, no. 2, p. 2, 2008.

[36] D. Chartash, "Maths meets myths: Quantitative approaches to ancient narratives," *Folklore*, vol. 130, pp. 315 – 317, 2019.

[37] J. Yose, *A network scientific approach to the quantitative analysis of epic texts*. Coventry University, 2017.

[38] J. Nescolarde-Selva, J. Usó-Doménech, and M. Lloret-Climent, "Mythical systems: mathematic and logical theory," *International Journal of General Systems*, vol. 44, no. 1, pp. 76–97, jan 2015.

[39] R. Kenna and P. MacCarron, "A networks approach to mythological epics," in *Maths Meets Myths: Quantitative Approaches to Ancient Narratives*. Springer, 2017, pp. 21–43.

[40] P. J. Miranda, M. S. Baptista, and S. E. de Souza Pinto, "The odyssey's mythological network," *PLoS One*, vol. 13, 2018.

[41] P. M. Carron and R. Kenna, "Universal properties of mythological networks," *EPL (Europhysics Letters)*, vol. 99, no. 2, p. 28002, jul 2012.

[42] ——, "Network analysis of the Íslendinga sögur – the Sagas of Icelanders," *The European Physical Journal B 2013 86:10*, vol. 86, no. 10, pp. 1–9, oct 2013.

[43] S. Akça and M. Akbulut, "Social network analysis of mythology field," *Library Hi Tech*, 2021.

[44] P. Mac Carron and R. Kenna, "Universal properties of mythological networks," *EPL (Europhysics Letters)*, vol. 99, no. 2, p. 28002, 2012.

[45] S. P. Borgatti, A. Mehra, D. J. Brass, and G. Labianca, "Network analysis in the social sciences," *Science*, vol. 323, no. 5916, pp. 892–895, 2009.

[46] J. Scott, "Social network analysis," *Sociology*, vol. 22, no. 1, pp. 109–127, 1988.

[47] S. Horvath, *Weighted network analysis: applications in genomics and systems biology*. Springer Science & Business Media, 2011.

[48] X. Zhu, Z. Duren, and W. H. Wong, "Modeling regulatory network topology improves genome-wide analyses of complex human traits," *Nat. Commun.*, vol. 12, no. 2851, pp. 1–15, May 2021.

[49] A.-L. Barabási, "Network science," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1987, p. 20120375, 2013.

[50] J. Fumanal-Idocin, A. Alonso-Betanzos, O. Cordón, H. Bustince, and M. Minárová, "Community detection and social network analysis based on the italian wars of the 15th century," *Future Generation Computer Systems*, vol. 113, pp. 25 – 40, 2020.

[51] S. Qazi, S. Ahmad, and K. Raza, "Using computational intelligence for tracking covid-19 outbreak in online social networks," in *Computational Intelligence Methods in COVID-19: Surveillance, Prevention, Prediction and Diagnosis*. Springer, 2021, pp. 47–59.

[52] S. Wasserman and K. Faust, *Social network analysis: Methods and applications*. Cambridge university press, 1994, vol. 8.

[53] E. Cambria, N. Howard, Y. Xia, and T.-S. Chua, "Computational intelligence for big social data analysis [guest editorial]," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 8–9, 2016.

[54] B. Wild, D. M. Dormagen, A. Zachariae, M. L. Smith, K. S. Traynor, D. Brockmann, I. D. Couzin, and T. Landgraf, "Social networks predict the life and death of honey bees," *Nat. Commun.*, vol. 12, no. 1, Feb 2021.

[55] G. Toth, J. Wachs, R. Di Clemente, A. Jakobi, B. Sagvari, J. Kertesz, and B. Lengyel, "Inequality is rising where social network segregation interacts with urban topology," *Nat. Commun.*, vol. 12, no. 1, Feb 2021.

[56] M. Gong, C. Song, C. Duan, L. Ma, and B. Shen, "An efficient memetic algorithm for influence maximization in social networks," *IEEE Computational Intelligence Magazine*, vol. 11, no. 3, pp. 22–33, 2016.

[57] C. S. Fischer, *To dwell among friends: Personal networks in town and city*. University of Chicago Press, 1982.

[58] A. Landherr, B. Friedl, and J. Heidemann, “A critical review of centrality measures in social networks,” *Business & Information Systems Engineering*, vol. 2, no. 6, pp. 371–385, 2010.

[59] N. Rescher, “On the status of “things in themselves” in Kant,” *Synthese*, vol. 47, no. 2, pp. 289–299, 1981.

[60] D. Jacquette, *Schopenhauer, philosophy and the arts*. Cambridge University Press, 2007.

[61] R. C. Cross, “Logos and forms in Plato,” *Mind*, vol. 63, no. 252, pp. 433–450, 1954.

[62] M.-C. Costa, A. Hertz, and M. Mittaz, “Bounds and heuristics for the shortest capacitated paths problem,” *Journal of Heuristics*, vol. 8, no. 4, pp. 449–465, 2002.

[63] A. V. Goldberg and C. Harrelson, “Computing the shortest path: A* search meets graph theory,” in *Proc. 16th ACM-SIAM Symposium on Discrete Algorithms, 2005*, vol. 5, 2005, pp. 156–165.

[64] E. Young, *Celtic Wonder-Tales*. Maunsel and Co., 1923.

[65] O. E. Coolidge, *Greek Myths*. Houghton Mifflin Company, 1948.

[66] S. Sturlson, *The Prose Edda*. New York, The American-Scandinavian Foundation, 1916.

[67] K. J. Wanner, *Snorri Sturluson and the Edda: the conversion of cultural capital in medieval Scandinavia*. University of Toronto Press, 2008, vol. 4.

[68] J. J. Webster and C. Kit, “Tokenization as the initial phase in NLP,” in *Proceedings of the 14th Conference on Computational Linguistics - Volume 4*, ser. COLING ’92. Association for Computational Linguistics, 1992, p. 1106–1110.

[69] C. Fox, “A Stop List for General Text,” *ACM SIGIR Forum*, vol. 24, no. 1-2, pp. 19–21, jan 1989.

[70] S. Bird, E. Klein, and E. Loper, *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc., 2009.

[71] B. Stroube, “Literary freedom: Project Gutenberg,” *XRDS: Crossroads, The ACM Magazine for Students*, vol. 10, no. 1, pp. 3–3, 2003.

[72] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian, “Forceatlas2, a continuous graph layout algorithm for handy network visualization designed for the gephi software,” *PloS One*, vol. 9, no. 6, p. e98679, 2014.

[73] M. Newman, *Networks*. Oxford university press, 2018.

[74] T. Ewing, “The birth of Lugh—Odin and Loki among the Celts,” *Sinsear*, vol. 8, p. 119, 1995.

[75] C. Lecouteux, *Phantom armies of the night: the wild hunt and the ghostly processions of the undead*. Simon and Schuster, 2011.

[76] E. Bakshy, I. Rosenn, C. Marlow, and L. Adamic, “The role of social networks in information diffusion,” in *Proceedings of the 21st international conference on World Wide Web*, 2012, pp. 519–528.

[77] W. Chen, L. V. Lakshmanan, and C. Castillo, “Information and influence propagation in social networks,” *Synthesis Lectures on Data Management*, vol. 5, no. 4, pp. 1–177, 2013.

6.10 The Krypteia ensemble: designing classifier ensembles using an ancient Spartan military tradition

Associated publication:

- Fumanal-Idocin, J., Cordón O., & Bustince, H. (2022). A The Krypteia ensemble: designing classifier ensembles using an ancient Spartan military tradition. *Information Fusion.*

  Status: Published.

  Impact Factor (JCR 2021): 17.564.

  Categories:

  Computer Science, Artificial Intelligence. Ranking 4/144 (Q1).

  Computer Science, Theory & Methods. Ranking 1/109 (Q1).

Information Fusion 90 (2023) 283–297

Contents lists available at ScienceDirect

Information Fusion

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

Full length article

The Krypteia ensemble: Designing classifier ensembles using an ancient Spartan military tradition

J. Fumanal-Idocin [a],[*], O. Cordón [b], H. Bustince [a]

[a] *Public University of Navarra and Institute of Smart Cities, Campus Arrosadia s/n, 31006 Pamplona, Spain*
[b] *Deptartment of Computer Science and Artificial Intelligence and Andalusian Research Institute DaSCI, "Data Science and Computational Intelligence", University of Granada, 18071 Granada, Spain*

ARTICLE INFO

Keywords:
Classifier ensemble
Optimal classifier selection
Social network
Human social behaviour
Multi-agent systems
Sparta
Krypteia

ABSTRACT

In this work we propose a new algorithm to train and optimize an ensemble of classifiers. We call this algorithm the Krypteia ensemble, based on an ancient Spartan tradition designed to convert their most promising individuals into future leaders of their society. We show how to adapt this ancient custom to optimize classifiers by generating different variations of the same task, each one offering different hardships according to distinct stochastic variables. This is thus applied to induce diversity in the set of individual weak learners. Then, we use a set of agents designed to select those subjects who excel in their assignments, and whose interaction minimizes excessive redundancies in the resulting population. We also study how different Krypteia ensembles can be stacked together, so that more complex classifiers can be built using the same procedure. Besides, we consider a wide range of different aggregation functions in the decision making phase to find the optimal performance for the different Krypteia ensemble variations tested. Finally, we study how different Krypteia ensembles perform for a wide range of classification datasets and we compare them with other state-of-the-art design techniques of classifier ensembles, obtaining favourable results to our proposal.

1. Introduction

Data analysis is one of the most popular disciplines in computer science in the present day [1,2]. There are many problems related to data processing that have been heavily studied due to their academic and economic interest, like data visualization [3], pattern discovery [4–6], and image processing [7] among others. One of the most common tasks in data analysis is to discriminate a series of inputs into a desired category, which is commonly called a classification task [8]. Some of the most popular classification algorithms are those based on neural networks [9], the family of Bayes classifiers [10,11], the K-Nearest neighbours [12], and the Support vector machines [13].

Due to the limitations of these models, a very popular approach to improve classification performance is to form an "ensemble" of classifiers [14–16], which consists of a set of classifiers trained under different configurations that make a decision together [17]. Usually, this consensus is formed by taking the average or the majority vote of these classifiers [18]. There are many classifier ensemble design approaches in the literature. E.g. the random forest trains a set of different decision trees under distinct subsampling conditions and then computes the final decision as the majority vote [19]; Bagging classifiers are formed by training many different subsamples with repeated samples from the original dataset [20]; and AdaBoost iteratively trains different classifiers, each one especially focused on the errors made by the previous classifiers [21]. Of course, since the development of these ensemble construction techniques there has been many research aimed at improving them [22–25]. Some of the most common techniques to improve the performance of an ensemble include using only a subset of the trained classifiers, which is called overproduce-and-choose [26], and using different kinds of classifiers, which is called an heterogeneous ensemble [27].

One of the most researched topics in classification ensembles is the decision making phase, where the majority vote or the arithmetic mean are substituted by another aggregation function [28]. Some of the most popular ones are the Choquet Integral [29], the Sugeno Integral [30], the Ordered Weighted Aggregation operators [31], and the Overlap functions [32]. It is also possible to use other aggregation functions obtained by means of the so-called penalty functions [33].

There are also some interesting research lines in classifier ensemble design focusing on the influence of diversity in the ensemble accuracy [34], and on the trade-off between accuracy and complexity (i.e. selecting the optimal number of classifiers) [35].

Bearing in mind the issues above, the objectives of this work are:

[*] Corresponding author.
E-mail addresses: javier.fumanal@unavarra.es (J. Fumanal-Idocin), ocordon@decsai.ugr.es (O. Cordón), bustince@unavarra.es (H. Bustince).

https://doi.org/10.1016/j.inffus.2022.09.021
Received 8 February 2022; Received in revised form 23 September 2022; Accepted 25 September 2022
Available online 30 September 2022

- To create a new algorithm to train an heterogeneous ensemble using stochastic conditions to adapt the original task to different difficulty settings.
- To study different hierarchical decision making schemes and different aggregation functions to obtain the final solution.

In order to do so, we propose a new approach to train an ensemble of classifiers based on an ancient Spartan ritual called the Krypteia [36]. In this ritual the young noble Spartans would be promoted to adulthood by proving themselves worthy through surviving alone in a hostile land for an unknown period of time. This situation made the participants act in situations of high uncertainty and lack or resources, which spurs heterodox thinking and smart use of resources. Due to the natural ever-changing conditions of real life, not two Krypteia rituals were the same, which also favoured diversity in the way subjects survived the trial, warranting individual fitness and compatible traits for the higher ranks of the Spartan society. This kind of training custom seems to tackle in real-life population some of the problems that are present in modern day classifier ensembles, such as ensemble diversity and accuracy [37], and the trade-off between them [38,39], or robustness against adversarial examples [40].

Following the same idea as in the original Krypteia, our proposed algorithm trains a heterogeneous ensemble using stochastic conditions to adapt the original task to different difficulty settings. Then, we use a novel technique to perform overproduce-and-choose to minimize redundancies in the system. We also study different hierarchical decision making schemes and different aggregation functions to perform the final solution.

Following this strategy, we expect the resulting population to learn different and intelligent solutions, according to each individual subject's situation, and to find good collective solutions when all the subjects' outputs are combined. The goodness of our proposal is shown in a large series of experiments in real-world datasets, comparing the results obtained using different Krypteia ensembles to other state-of-the-art classifier ensemble design methods.

The rest of this paper goes as follows: first, in Section 2 we discuss some relevant works in ensemble design. In Section 3 we recall some concepts of aggregation functions. In Section 4 we explain the ancient ritual of the Krypteia and our proposal to emulate it in a computational environment. Next, in Section 5 we detail the specifics of each step in the design process. Then, in Section 6 we describe the experiments we performed using Krypteia ensembles, and in Section 7 we compare the obtained results to those generated using other types of ensembles. Finally, in Section 9 we give our final remarks and future lines for this work.

2. Related work

Due to their massive popularity and numerous applications, many works have been devoted to enhance performance in classifier ensemble systems. Some of the main lines of research in this topic are model generation for the classifier, model selection for the classifier, and combining the output from the classifiers.

2.1. Individual classifier generation and diversity induction

Ensemble generation is based on learning individual classifiers, weak learners, whose outputs can be combined then into one final output for the global system [41]. The ideal set of classifiers for an ensemble are both accurate and complementary, so that the errors committed by the individual models are corrected in the collective decisions taken.

The most popular ways to generate classifiers are bagging [20], in which each model is trained using a random subsample from the original data; boosting [42], where classifiers are iteratively trained based on the previous errors obtained by the ensemble; and clustering-based approaches [43], where data are clustered according to the different patterns found in the original data and a dedicated classifier is used to classify the samples for each cluster. From this classical approaches, bagging is usually preferred to boosting, as bagging can be computed in a parallelized setting, while boosting requires an iterative process. Clustering based approaches can be very effective when the decision boundaries in the dataset are not constant. However, they are limited by the structures found by the clustering algorithm used. For example, if we used the popular K-Means algorithm, we can only detect convex structures, so complex regions will still be problematic.

Depending on the kind of classifiers used, we denote a homogeneous ensemble, if all the classifiers are of the same type, and a heterogeneous ensemble, if they are different. Many approaches have been studied in both paradigms. In [44] the authors proposed a homogeneous ensemble of neural networks for word classification and a heterogeneous one was used in [45] for a similar problem, using deep learning and classical algorithms as well. Also, in [46] the authors studied heterogeneous ensembles applied to online data streams. Heterogeneous ensembles offer more diversity than their homogeneous counterpart. Nevertheless, the proportions in which the different classifiers should form the ensemble design involves another problem [47]. In [48] the performance of heterogeneous ensembles is compared to homogeneous ones to deal with imbalanced classification problems, finding favourable results to the former ones.

2.2. Classifier selection

As the usefulness of each of the trained classifiers can vary significantly [49], one popular approach in ensemble design is to choose only a subset of classifiers, or to purge a percentage of the classifiers generated [50], which is commonly called overproduce-and-choose (OCS) [26].

Recent works regarding classifier selection and pruning include the proposal in [51] where the authors prune a pool of ensembles based on the indecision region of each classifier. Meanwhile, the proposal in [52] uses the K-means algorithm to cluster the candidates and find the ones that minimize redundancies. Another successful approach to purge classifiers is using meta-features [53]. Meta-features are features extracted from the original data that are used to train meta-classifiers that discriminate between good and bad candidates. Selecting classifiers can be useful when there is a lot of redundancy in their outputs. However, this redundancy can sometimes be useful to minimize the impact of outlier predictions and underperforming classifiers. Meta-feature methods can also tackle this problem by determining which classifiers are good for each sample. However, they also impose additional design problems, like determining which meta-features are good for this task and the boundary conditions to determine if a classifier is competent or not.

Finally, it is also possible to use optimization algorithms to choose the most effective subset of classifiers [54,55] or features [56]. Optimization approaches can lead to good results both in the case of classifier and feature selection. However, they require a proper modelling and a suitable algorithm for this task. Besides, as the validation set used to optimize each configuration is usually only a fraction of the training set, there is also the risk of overfitting.

2.3. Classifier combination

The classifier fusion is commonly performed using functions such as the maximum, the arithmetic mean, and the majority vote [24,57,58]. These functions work best when there is independence between the errors among the classifiers. However, that condition is usually not guaranteed [38].

A common solution to this problem is to ponder each classifier according to its importance [59]. Other relevant approaches include the use of fuzzy aggregations to model uncertainty and coalitions of the inputs to fuse. Some popular operators in this phase include the Choquet and Sugeno integrals [60–62]. These operators can learn the interaction among the features and take them into account when aggregating the data. They are also simpler to use compared to other pruning or feature selection mechanisms. However, they also require to learn a suitable fuzzy measure to capture these coalitions, which is not straightforward [63]. Another successful strategy consist of using fuzzy linguistic rule-based classification system as the fusion process, so that it can be interpretable for the user [64].

It is also possible to fuse the classifiers in different phases, using a hierarchical fusion phase [18,27], which can also include different aggregation operators [60]. This procedure can achieve higher accuracy results than fusing all the classifiers in one phase. However, it requires a sensitive hierarchy system among the classifiers and a suitable aggregation operator in each phase. Choosing the best aggregation operator for one vector can be solved using a penalty or a moderate deviation [65], but the interaction among aggregations in a hierarchical fashion has not been studied.

3. Preliminaries

In this section we shall recall some of the concepts related to the most common aggregation functions, the formulation of the Choquet and Sugeno integrals, and some of their generalizations, the Overlap functions and the Ordered Weighted Aggregation operators.

3.1. Properties of aggregation functions

Aggregation functions are used to fuse information from n sources into one single output. A function A: $[0,1]^n \to [0,1]$ is said to be a n-ary aggregation function if the following conditions hold:

- A is increasing in each argument: $\forall i \in \{1,\dots,n\}$, if $x_i < x_j$, $A(x_1,\dots.,x_i,\dots x_n) \leq A(x_1,\dots,x_j,\dots x_n)$. For example, consider the vectors $\mathbf{x} = \{0.3, 0.9, 0.1, 0.8\}$ and $\mathbf{z} = \{0.3, 0.9, 0.6, 0.8\}$. If A is an aggregation function, it must hold that $A(\mathbf{x}) \leq A(\mathbf{z})$, because all elements of in $\mathbf{x}$ are equal than those in $\mathbf{z}$, except for the case of x_3 and z_3, where $0.1 < 0.6$.
- $A(0,\dots,0) = 0$
- $A(1,\dots,1) = 1$

Some examples of classical n-ary aggregation functions are:

- Arithmetic mean: $A(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i$.
- Max: $A(\mathbf{x}) = max(x_1,\dots,x_n)$.
- Min: $A(\mathbf{x}) = min(x_1,\dots,x_n)$.

3.2. T-norm

A T-norm is an aggregation function $[0,1]^2 \to [0,1]$ that satisfies the following properties for $x, y, z \in [0,1]$ [66]:

- $T(x,y) = T(y,x)$
- $T(x,T(y,z)) = T(T(x,y),z)$
- $T(x,1) = x$

Some examples of T-norms are the product, the minimum and the Łukasiewicz T-norm:

$$L_{luk}(x,y) = \max(0, x+y-1) \tag{1}$$

3.2.1. Overlap functions

An n-dimensional overlap is an aggregation function $G : [0,1]^n \to [0,1]$ such that [32]:

- G is commutative.
- $\prod_{i=1} x_i = 0$ if and only if $G(\mathbf{x}) = 0$.
- $\prod_{i=1} x_i = 1$ if and only if $G(\mathbf{x}) = 1$
- G is increasing.
- G is continuous.

Some examples of overlap functions are:

- Minimum: $G(\mathbf{x}) = \min(\mathbf{x})$
- Harmonic Mean (HM): $G(\mathbf{x}) = \frac{n}{\sum_{i=1}^{n}\frac{1}{x_i}}$
- Sinus Overlap (SO): $G(\mathbf{x}) = \sin(\frac{\pi}{2}\mathit{\Pi}_{i=1}^{n} x_i)$
- Geometric Mean (GM): $G(\mathbf{x}) = \sqrt[n]{\prod x_i}$

3.3. Ordered Weighted Averaging operators (OWA)

$\mathbf{w} = (w_1,\dots,w_n) \in [0,1]^n$ is called a weighting vector if $\sum_{i=1}^{n} w_i = 1$. The OWA operator associated to $\mathbf{w}$ is the mapping $\text{OWA}_{\mathbf{w}} : [0,1]^n \to [0,1]$ defined for every $\mathbf{x} = (x_1,\dots,x_n) \in [0,1]^n$ by [67]:

$$OWA(\mathbf{x}) = w_1 x_{\gamma(1)} + \dots + w_n x_{\gamma(n)} \tag{2}$$

where $\gamma : \{1,\dots,n\} \to \{1,\dots,n\}$ is a permutation such that: $x_{\gamma(1)} \geq x_{\gamma(2)} \geq \dots \geq x_{\gamma(n)}$.

The weighting vector can be computed used a quantifier function, Q. For this study, we have used the following one:

$$w_i = Q_{a,b}(\frac{i}{n}) - Q_{a,b}(\frac{i-1}{n}) \tag{3}$$

$$Q_{a,b}(x) = \begin{cases} 0, & \text{if } x < a \\ 1, & \text{if } x > b \\ \frac{x-a}{b-a}, & \text{otherwise} \end{cases} \tag{4}$$

where $a, b \in [0,1]$ and $a < b$. Depending on the value of the parameters a and b, different weight vectors can be obtained. We have considered three different ones:

- OWA_1: $a = 0.1, b = 0.5$
- OWA_2: $a = 0.5, b = 1$
- OWA_3: $a = 0.3, b = 0.8$

3.4. Choquet integral

Having $N = \{1,\dots,n\}$, a function $m : 2^N \to [0,1]$ is a fuzzy measure if, for all $X, Y \subseteq N$, it satisfies the following properties [28]:

($m1$) Increasingness: if $X \subseteq Y$, then $m(X) \leq m(Y)$.
($m2$) Boundary conditions: $m(\emptyset) = 0$ and $m(N) = 1$.

The discrete Choquet integral with respect to m is defined as the function $C_m : [0,1]^n \to [0,1]$ given for every $\mathbf{x} = (x_1,\dots,x_n) \in [0,1]^n$, by:

$$C_m(\mathbf{x}) = \sum_{i=1}^{n}(x_{\sigma(i)} - x_{\sigma(i-1)}) \cdot m(A_i) \tag{5}$$

where $\mathbf{x}_\sigma$ is an increasing permutation of $\mathbf{x}$ such that $0 \leq x_{\sigma(1)} \leq \dots \leq x_{\sigma(n)}$. With the convention that $x_{\sigma(0)} = 0$, and $A_i = \{(i),(i+1),\dots,(n)\}$.

Two important generalizations of the Choquet integral are the CF [61] the $\text{C}_{F1,F2}$ [68] integrals, and the d-Choquet integrals, in which the difference between the inputs in Eq. (5) is changed by a dissimilarity [69].

3.4.1. CF integral

The CF is a generalization of the Choquet integral that replaces the product used in Eq. (5) for a more general function F. In [70] the authors detail the required properties for F so that the CF is an aggregation or a pre-aggregation function, and conclude that the best F in their experimental results is the Hamacher T-norm. For this reason, we have chosen it for our experimentation, as detailed in the following expressions:

$$T_H(x,y) = \begin{cases} 0, & \text{if } x = y = 0 \\ \frac{xy}{x+y-xy}, & \text{otherwise} \end{cases} \tag{6}$$

$$CF_m(\mathbf{x}) = \sum_{i=1}^{n} T_H(x_{\sigma(i)} - x_{\sigma(i-1)}, m(A_i)) \tag{7}$$

3.4.2. $C_{F1,F2}$ integral

The original product of the Choquet Integral can be decomposed on two product functions using the distributive property of the product. Therefore, the Choquet integral can be written as:

$$C_m(\mathbf{x}) = \sum_{i=1}^{n} x_{\sigma(i)}m(A_i) - x_{\sigma(i-1)}m(A_i) \tag{8}$$

Then, the product functions are substituted for two more generic functions: F_1 and F_2. In [68] the authors explain the properties that must hold F_1 and F_2 so that the $C_{F1,F2}$ is an aggregation or a pre-aggregation function. Consequently, the expression for the $C_{F1,F2}$ is the following:

$$C_{F1,F2}(\mathbf{x}) = \sum_{i=1}^{n} F_1((x_{\sigma(i)}), m(A_i)) - F_2((x_{\sigma(i-1)}), m(A_i)) \tag{9}$$

For our experimentation, we take $F_1 = \sqrt{xy}$ and F_2 is the Łukasiewicz T-norm.

3.5. Sugeno integral

Let $m : 2^N \to [0,1]$ be a fuzzy measure. The discrete Sugeno integral with respect to m is defined as a function $S_m : [0,1]^n \to [0,1]$, given for every $\mathbf{x} = (x_1, \ldots, x_n)$ [71], following the same notation as in the Choquet integral in Eq. (5):

$$S_m(\mathbf{x}) = \max\{min(x_{\sigma(i)}, m(A_i)) | i = 1, \ldots, n\} \tag{10}$$

Two generalizations of the Sugeno Integral are the Hamacher-based Sugeno integral and the FG-Sugeno.

3.5.1. Hamacher-based Sugeno integral

If we consider using the Hamacher T-norm instead of the minimum in Eq. (10), we obtain the following expression [72]:

$$S_m^{TH}(\mathbf{x}) = \max\{T_H(x_{\sigma(i)}, m(A_i)) | i = 1, \ldots, n\} \tag{11}$$

3.5.2. FG-Sugeno

If we replace the minimum for the product, and the maximum for the sum in Eq. (10), we obtain the following expression [62]:

$$S_m^{FG}(\mathbf{x}) = \sum_{i=1}^{n} (x_{\sigma(i)}m(A_i)) \tag{12}$$

4. The Krypteia ensemble

The Krypteia ensemble is a novel classifier ensemble algorithm designed to maximize the effectiveness of each individual subject and the variability and performance of their outputs when combined. It does so by mimicking the ancient rite of Krypteia in the ancient Sparta, designed to train the future elites of the Spartan army and government.

This algorithm consists of three main steps:

1. Survival ordeal: in this phase we train each classifier individually. In order to induce diversity in the training process, we modify the training task for each one in a stochastic process, so that some of them have a easier or harder task than the original classification task. We discard all the classifiers that did not meet the expected accuracy rate in their own task.
2. Social ordeal: this second phase follows an OCS scheme in which we discard those classifiers that have a very similar output to other classifiers with higher accuracy rate. The aim is to avoid having samples with much more weak learners classifying them correctly than others.
3. Aggregation learning: we learn which is the most appropriate function to combine the output of all the surviving classifiers.

The resulting population from this process is called a Krypteia Unit.

A general scheme of the Krypteia algorithm is displayed in Fig. 1. The following subsection briefly describes the Krypteia ritual in ancient Sparta design method and Section 4.2 describes each step of the Krypteia ensemble in detail.

4.1. Krypteia in the ancient Sparta

The Krypteia was a particularly brutal initiation rite in the Spartan society for the young men of the higher ranks of the state [36]. According to Plutarch [73], each year the young noblemen of Sparta would declare war to the Helot population of Sparta, so that any killing or robbery committed was not considered crime. Armed with nothing but a knife, the young Spartans were left alone and sent out in the night to the Helot settlements. They were supposed to obtain their own methods of survival by stealing and killing in their circumvent area.

The origins and exact purpose of the Krypteia are still under debate. This ritual was supposed to be a form of terrorizing and subjugating the Helot population, alongside training the next generation of Spartan leaders, as no young man could aspire to hold positions of power if he had not passed through this ordeal. It is also believed that the Krypteia participants could have been organized as a unit in the Spartan army. In any case, such brutal practices seemed to be effective, as Sparta made a place for itself in history thanks to is great military capacity, and as long as the Krypteia took place, the Helot population stayed under their rule.

It is believed that the Krypteia was disbanded in the battle of Sellasia, in 222 BC, where Sparta lost against the Macedonian army commanded by Antigonus III [74]. This resulted in the emancipation of many helots and without a Helot population, it was impossible to organize the Krypteia.

4.2. Krypteia ensemble: bringing the ancient ritual to modern computational systems

In this section we detail both the Survival and Social ordeals, and how to combine their outputs.

4.2.1. Survival ordeal (Fig. 1-Step 2)

The Survival ordeal is a diversity induction process that consists of randomly modifying the original task in order to obtain a different version of the weak learner. The aim is that this new task should be more difficult than the original most of the times, so that the resulting Krypteia Unit will have classifiers adapted to harder or extreme situations compared to the original, single classifier.

We alter the tasks using two different techniques: sampling modifications, denoted as “Data ordeal”, and prediction alterations, which we called “Bias ordeal”. The Data ordeal consists of performing an undersampling, data augmentation, or oversampling technique of one or various classes in the classification task. The Bias ordeal consists of adding an artificial bias to one or more classes to the actual output of the classifier. Whether or not a classifier is affected by the Data ordeal or the Bias ordeal, the extent of that ordeal and the number of classes affected, is all decided randomly.

Once the Data ordeal and Bias ordeal have been completed we evaluate the performance of the altered classifier. If this performance is less than a survival threshold, we discard that classifier. Following the Krypteia metaphor, this means that the soldier was not strong enough to survive the ritual and was killed by the Helots or the environment.

The ordeals can be too hard, so that all classifiers fail. It might also happen that the number of surviving classifiers is higher than the expected value. So, it is important to take into consideration the natural difficulty of the original classification task in order to establish the survival threshold, or to establish additional measures to guarantee that a reasonable number of weak learners survive the Survival ordeal.

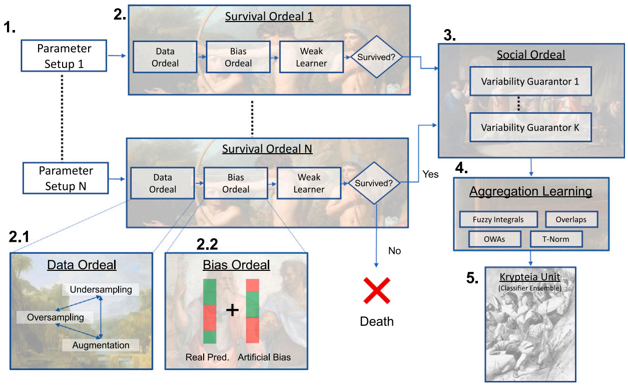


Fig. 1. Visual scheme of the Krypteia algorithm. **1.** We generate random sets of parameters. **2.** Each of these parameter settings creates a different Survival ordeal, where the weak learners need to correctly solve a stochastically modified version of the original classification task. The modification includes the Data ordeal (**2.1**), where we manipulate the training data, and the Bias ordeal (**2.2**), where we compute a random vector that we will add as an artificial bias to each of the classifiers predictions. **3.** If they pass the Survival ordeal, a selection of the surviving classifiers is performed by k Variability Guarantors in the Social ordeal to minimize redundancies. **4.** We choose the best aggregation function for the decision making phase of the ensemble. **5.** The output of the Social ordeal is the Krypteia Unit. Section 4.2 contains a detailed description of each component behaviour.

4.2.2. Social ordeal (Fig. 1-Step 3)

Once the individual classifiers have been proved by the Survival ordeal, we make the population of classifiers pass through the Social ordeal, which is an OCS scheme that measures how diverse is the "society" formed by these classifiers and removes those that produce redundant results with respect to the rest of the population.

The idea is that, although the variability within the Survival ordeal itself makes the classifiers different, there could be redundancies, especially if the survival rate is high. For example, given 0 as a complete failure and 1 as a total success in the Survival ordeal, if we set the survival threshold to 0.95, then we are forcing all the surviving classifiers to present very similar outputs.

To solve this problem, in the Social ordeal we create a set of n Variability Guarantors (VGs), whose aim is to guarantee the diversity in the output of each classifier in the final Krypteia Unit. To do so, each VG subsamples a very small fraction of the input data for each subject (1%, for example) and then checks how each classifier performs for this data split.

Each VG marks as "good" the subjects that correctly classified its subsampled data. Since each VG takes only a small fraction of the input data, the intersection of the data chosen among the VGs will be virtually null. In this way, we avoid having in the final Krypteia Unit a long list of classifiers that performed very well in the same subset of data, whilst performed very poorly on other. Besides, since the VGs uniformly sample the dataset, their subsampled data can contain both easy and hard to correctly classify observations. In this way, VGs reject those classifiers that performed well only on the easy set of observations, even if they obtained a higher accuracy rate than the classifiers marked as good by the VGs.

4.2.3. Decision making

The most straightforward way to make a decision with a Krypteia Unit is to fuse the output of each weak learner using any of the aggregation functions presented in Section 3. However, although a Krypteia Unit can obtain good results on its own, the stochastic nature of the training process can result in many different outcomes. This is one of the strengths of the Krypteia, but it can also result in poor performance in some cases. To minimize the negative impact of the possible faulty units, we propose to stack different Krypteia Units in different decision making phases, in a hierarchical decision making scheme. This proposal has another benefit: we can use different aggregation functions in each level, which can result in better performance.

The different hierarchical schemes that can be used are illustrated in Fig. 2. A decision from a set of N Krypteia units can be obtained in three different ways:

1. Unit-all: we fuse the output from all weak learners within the Krypteia units in one phase, using just one aggregation function.
2. Division-all: we denote as a Krypteia Division the appending of the output of different Krypteia Units. Then, the Division-all scheme consists of fusing the output of various Krypteia Divisions obtained from the N Krypteia units, using one aggregation function in each Krypteia Unit and another for the Krypteia divisions. Krypteia units are assigned to different Krypteia divisions randomly.
3. Krypteia army: consists of obtaining the output for every Krypteia Division individually and then fusing their outputs, using a total of three aggregation functions.

5. Training a Krypteia Unit for a classification problem

In this section we focus on how to implement the following stages of a Krypteia Unit training process:

1. How to setup the parameters.
2. How to perform the Survival ordeal, and how difficult it should be.
3. How to configure the Social ordeal.
4. How to choose the aggregation function to make the decision.

5.1. Parameter setup

For the case of a classification system, a Krypteia consists of a set of weak learners. However, we do not specify each one individually; instead, we set a list of parameters for the whole Krypteia Unit (see Table 1 for example) in order to induce diversity within each unit. These parameters are:

1. The number of subjects wanted in the Krypteia Unit.
2. The proportion of each type of classifiers.

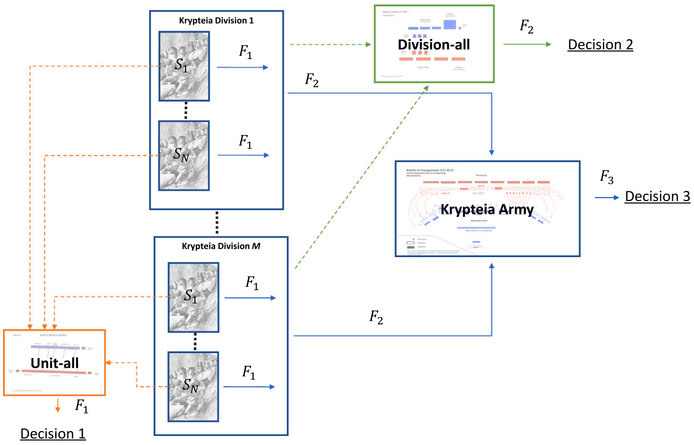


Fig. 2. Visual scheme showing the algorithm proposed in Section 4.2.3 to generate and combine the output of multiple Krypteia Units. A Krypteia Division is obtained by appending the output from N Krypteia Units and a Krypteia Army by appending and fusing the output of M Krypteia Divisions. A decision from a set of N Krypteia Units can be obtained in three different ways: **Decision 1**: consists of fusing all the weak learners within the Krypteia Units in one phase, using just one aggregation function. This scheme is denoted "Unit-all". **Decision 2**: consists of appending the output of the Krypteia Divisions, using one aggregation function to fuse each Krypteia Unit and another for the Divisions. This scheme is called "Division-all". **Decision 3**: consists of fusing every Krypteia Division individually and the fusing the output from all of them, using a total of three different aggregation functions. This scheme is the Krypteia Army. Images taken from [75].

Table 1
Example of two Krypteia Unit configurations. For each unit, we specify the number of classifiers wanted in the final population and the proportion of each kind, the number of VGs, the minimum accuracy needed to pass the Survival ordeal, and the ordeals performed in this unit.

N	KNN	LDA	QDA	SVM	Tree	VGs	Survival Thr.	Ordeal
24	0.08	0.12	0.20	0.12	0.25	6	0.5067	Survival and Social
7	0.14	0.14	0.14	0.28	0.28	–	0.9266	Survival

3. The number of VGs in the Social ordeal.
4. The initial Survival Threshold.
5. Which Ordeals to perform.

By setting different parameters for each unit, we can maximize the diversity between various Krypteia Units trained for the same task. This can be of use, for example, if we want to train different Krypteia Units and then choose the best one, or if we want to use the Krypteia Division or a Krypteia Army decision schemes.

For our tests, we have worked with five different types of classifiers:

- K-Nearest Neighbours (KNN) (K = 6 for this experimentation) [12].
- Linear Discriminant Analysis (LDA) [76].
- Quadratic Discriminant Analysis (QDA) [77].
- Support Vector Machines (SVM) [13].
- Decision trees [78].

We choose a random proportion for each type of classifier, a random number between 1 and 10 for the number of VGs, and a random number between 0 and 1 for the Survival Threshold. The type of Ordeal can be: Survival only, Social only, or both, each of the possibilities with the same probability.

5.2. Designing the Survival ordeal

The Survival ordeal consists of two complementary ordeals: the Data ordeal and the Bias ordeal, in which we artificially modify the weak learner behaviour. Once both ordeals have been stated, we also need to adjust the Survival Threshold so that the survival rate is within acceptable boundaries.

5.2.1. Data ordeal

The Data ordeal consists of modifying the training data for a weak learner. We can do so by subsampling the data, which theoretically makes the problem harder, by using oversampling techniques, which should make the problem easier for the classifier, or by using both. In that case we can also differ in which order both procedures are applied. The idea is not to necessarily obtain a stratified dataset, which is not guaranteed to happen, but to obtain maximum variation in the different synthetic training sets generated.

We have used the following techniques to perform the sampling manipulations:

- Random undersampling: which consists of randomly sampling a percentage of the original data.
- Centroid Undersampling (CU): we use the K-means algorithm to compute k centroids, being k the number of samples wanted from the original data.
- Synthetic Minority Over-sampling TEchnique [79] (SMOTE): generates new samples by interpolating examples from the same class.
- SMOTE + Tomek Links (SMOTETomek) [80]: combines the oversampling of SMOTE and Tomek Links to delete overlapping observations between classes.
- SMOTE + Wilson's Edited Nearest Neighbour Rule (SMOTEEN) [80]: combines the oversampling of SMOTE with undersampling based on the Edited Nearest Neighbour Rule [81].

In order to determine how much each event is going to happen, we set a parameter p. The lower this p, the more likely is that a data manipulation happens.

The scheme the followed by the Data ordeal is:

1. Draw a random number. If this number is bigger than p, then the Data ordeal takes place (step 2); if not, then there are no data manipulations (step 3).

2. Draw a random number. If this number is bigger than 0.5, then we proceed to subsample the data and then oversample it (step 2.1). If not, first we perform the oversampling and then undersampling (step 2.2).

    2.1. Draw a random number. If this number is bigger than p, then we perform the undersampling (Alg. 2). Draw a random number again, and if is bigger than p, we perform the oversampling (Alg. 1).

    2.2. Draw a random number. If this number is bigger than p, then we perform the oversampling (Alg. 1). Draw a random number again, and if is bigger than p, we perform the undersampling (Alg. 2).

3. We train the weak learner on the resulting data.

Algorithm 1: Generating Data Advantage in the Data ordeal

```
Result: $new\_data$
Input: $train\_data$, $p$
$lucky\_number = random()$
if $lucky\_number > p$ then
    if $lucky\_number < p + 1/3 * (1 - p)$ then
        $new\_data = SMOTETomek(train\_data)$
    else if $lucky\_number < p + 2/3 * (1 - p)$ then
        $new\_data = SMOTEEN(train\_data)$
    else
        $new\_data = SMOTE(train\_data)$
    end
end
```

Algorithm 2: Undersampling in the Data ordeal

```
Result: $new\_data$
Input: $train\_data$, $p$
$lucky\_number = random()$
if $lucky\_number > p$ then
    if $lucky\_number < p + 1/2 * (1 - p)$ then
        $new\_data = random\_undersample(train\_data)$
    else
        $new\_data = centroid\_undersample(train\_data)$
    end
end
```

5.2.2. Bias ordeal

The Bias ordeal consists of inducing an artificial bias in the weak learners predictions. The process follows a similar procure to the Data ordeal. We draw a random number, if that number is higher than p, the Bias ordeal takes place, if not, no bias is induced.

To perform the bias induction we draw C random numbers from a uniform distribution in the $[0, 1]$ interval, where C is the number of classes in the classification task, obtaining the vector r_C. When the classifier predicts the probability for each class for a sample, we add the r_C vector to those probabilities.

Although the aim of this ordeal is to augment the variance among weak learners by reducing the individual accuracy of each one, we also have found cases where this artificial bias also improved the performance of individual weak learners. This might be because under the right circumstances, like a very imbalanced training set, this artificial bias can work as a regularization factor.

5.2.3. Setting the Survival Threshold

The Survival Threshold quantifies the level of exigence that we impose to the weak learners after they passed their Survival ordeal in order to accept them as valid weak learners. It is expressed as the minimum accuracy rate required to obtain in the original training set after passing the Data and Bias ordeals. It is a very important factor for the final Krypteia Unit: if it is too low, the resulting classifiers can output almost random predictions; if it is too hard, no classifier will come out. However, this value cannot be interpreted as the ordeal difficulty or the percentage of survival rate. A Survival Threshold of 0.9 can be too easy for a classification task that any individual classifier can perform almost perfectly, and a Survival Threshold of 0.5 can be too much if the learning task is very difficult.

The solution to this problem is an adaptative threshold, i.e to start with a random Survival Threshold, and then decrease it if the death rate is too high. We reduce the Survival Threshold by 0.05 each time we register as many deaths as twice the number of classifiers in the Unit until the desired number of classifiers is obtained. That mimics the robust approaches followed by metaheuristics to adapt their main diversification–intensification trade-off parameter [82].

5.3. Configuring the Social ordeal

The VGs compose the Social ordeal. They are supposed to convert a pack of good individual classifiers into a truly functional ensemble of classifiers. Two parameters are important to form a collective of VGs that perform a meaningful Social ordeal:

- VG exigence: this is the percentage of the original data that the VG will use to evaluate each weak learner. The lower this parameter is, the more strict each VG will be. This might present a problem if the VGs are too strict, because if the number of test samples is too low, then many of the classifiers might get all of them right, or on the contrary, fail them all. If we choose a value that is too high, then the VGs will have a significant intersection among them, which would also make the diversity search futile. We have chosen value 1% as it seems to correctly balance both issues and avoids oversaturation of one VG.
- Number of VGs: the more VGs, the more subjects will be marked as “good”. We have opted to draw a distribution between 1 and 10, so that with a exigence of 1%, the VGs will sample around $1-10\%$ of the original data.

5.4. Choosing the aggregation function

Once all the subjects have passed through all the ordeals, the final population forms a Krypteia Unit ensemble. This ensemble of classifiers needs to fuse the output of each individual weak learner to reach a final decision. We have studied the functions mentioned in Section 3.1 as possible aggregation functions for the Krypteia.

In order to discriminate the best aggregation function, we have opted for performing a 5-fold validation in the training set for each one of them, and choose the aggregation that resulted in a best accuracy according to this evaluation criteria. In the case of the Krypteia Division and the Krypteia Army, we have tested using all possible combinations of aggregation functions in each phase.

6. Experimental results for different Krypteia ensembles

In this section we study the performance of the Krypteia ensemble for a wide range of classification datasets. We show the performance for different decision making strategies, a study of the importance of the different parameters of a Krypteia Unit, and how the Krypteia ensembles compare to other kinds of classifier ensemble design approaches.

6.1. Datasets studied in the experimentation

For our experimentation, we used a large set of 52 different classification datasets, obtained from the Keel database [83]. The number of samples and attributes for each one are reported in Table 2. We take a standard 80/20 training-test partition for each dataset. The metric used to measure the performance is the standard classification accuracy.

Table 2
Datasets used for our experimental results.

Dataset	Instances	Features	Classes	Dataset	Instances	Features	Classes
abalone	4173	8	28	nursery	12 959	8	5
appendicitis	105	7	2	optdigits	5619	64	10
australian	689	14	2	page-blocks	5471	10	5
balance	624	4	3	penbased	10 991	16	10
banana	5299	2	2	phoneme	5403	5	2
bands	364	19	2	pima	767	8	2
breast	276	9	2	post-operative	86	8	3
bupa	344	6	2	ring	7399	20	2
car	1727	6	4	saheart	461	9	2
chess	3195	36	2	satimage	6434	36	6
coil2000	9821	85	2	segment	2309	19	7
contraceptive	1472	9	3	sonar	207	60	2
crx	652	15	2	spambase	4596	57	2
dermatology	357	34	6	spectfheart	266	44	2
ecoli	335	7	8	splice	3189	60	3
flare	1065	11	6	texture	5499	40	11
german	999	20	2	thyroid	7199	21	3
haberman	305	3	2	tic-tac-toe	957	9	2
housevotes	231	16	2	titanic	2200	3	2
ionosphere	350	33	2	twonorm	7399	20	2
iris	149	4	3	vehicle	845	18	4
letter	19 999	16	26	vowel	989	13	11
magic	19 019	10	2	wdbc	568	30	2
mammographic	829	5	2	wine	177	13	3
mushroom	5643	22	2	wisconsin	682	9	2
newthyroid	214	5	3	yeast	1483	8	10

6.2. Results for the Krypteia Unit

In this section we have studied the results using 300 Krypteia Units for each dataset, trained using the procedures in Section 5. We chose that number to honour the famous Battle of Thermopylae between Spartans and Persians [84].

Table 3 collects the results for the Krypteia Units in each one of the datasets considered, divided into three columns. In the first column, we show the average of all the Krypteia Units' performance. In the second column, we show the results that the best Krypteia Unit obtained (the one with better average accuracy for all the datasets), and finally in the third column we show the best results obtained by a Krypteia Unit in that particular dataset.

In most of them, the difference between the average result and the best Krypteia Unit is not significant, which shows that the Krypteia training process is indeed capable of generally generate good results. However, in some datasets, for example in "letter", there is a +30% accuracy difference between the average and the best unit. This shows that the Krypteia training process is also capable to create vastly superior subjects than the average Krypteia Unit.

In Table 4 we show the average for each column in Table 3, in order to obtain the average performance for all Krypteia Units, the average performance for the best Krypteia Unit, and the average best result over all the datasets considered. We can see here that the best unit generated is clearly superior to the average performance of the rest of the units, and that there is still an average of 2% of difference between the optimal result and the best unit obtained.

6.3. Feature importance in the Krypteia ensemble

As many parameters as the Krypteia Unit has, it is natural to think that some of them are more important than others. In order to compute the importance of each feature, we built a classification and regression tree (CART). The CART model have been used widely in medicine as a way to measure the effects of different treatments in the outcome of a patient. This algorithm trains a random forest with the parameters of a model as inputs and the final outcome of the model as labels [85]. It predicts the performance of a model based on its parameters. Then, we can study the coefficients that the CART model applied to each factor to make its prediction, to learn how they affect the performance of the studied system.

In Table 5 we displayed the results for this analysis, illustrating the Krypteia Units performance. From this table we can infer that the most important factor is the Survival Threshold by a large margin, followed by the Unit size. The proportion of different classifiers seems to be equally important, although using trees seemed to be less important than the rest. The kind of ordeal performed seems not to have much effect.

6.4. Results for the Krypteia Division

In this case, we used the same Krypteia Units as in Section 6.2, but instead of studying the performance of each individual Spartan unit, we followed the process detailed in Section 4.2.3 to stack different Krypteia Units. Table 6 collects the results for each dataset for the average of all Krypteia Divisions, the best Krypteia Division, and the best result obtained by a Krypteia Division for each one.

In Table 7 we show the average of all columns in Table 6. In this case, we obtained a lesser difference between the best and the average result compared to the unit results. This seems to indicate that stacking the units in this way mitigates the impact of bad units, although the best possible result is inferior to that obtained using Krypteia Units.

6.5. Results for the Krypteia Army

In this case we have stacked the Krypteia Units forming one Krypteia Army. We used the same Krypteia Divisions as in Section 6.4 in order to directly compare the effect of an additional level of complexity.

We display the results obtained with the Krypteia Army in Table 8 and the correspondent aggregation trio that achieved that result. We can observe that even though the arithmetic mean is most of the times the chosen aggregation, studying additional aggregation functions allows in many cases to improve the result.

6.6. Comparison of the different Krypteia ensembles

In Table 9 we show the comparison for the average performance of the different Krypteia ensembles: the best Krypteia Unit generated, the best Krypteia Division generated, and the Krypteia Army. We found

Table 3
Results for the Krypteia Units.

Dataset	Avg.	Best unit	Best Res.	Dataset	Avg.	Best unit	Best Res.
abalone	99.73	100.00	100.00	nursery	89.83	99.96	100.00
appendicitis	94.67	95.24	100.00	optdigits	93.80	93.51	98.84
australian	81.57	87.68	88.41	page-blocks	38.61	96.99	97.44
balance	99.78	100.00	100.00	penbased	87.53	97.31	99.49
banana	88.02	90.09	90.47	phoneme	88.54	88.90	90.56
bands	99.34	94.52	100.00	pima	99.64	100.00	100.00
breast	99.44	100.00	100.00	post-operative	39.69	100.00	100.00
bupa	67.35	68.12	78.26	ring	91.11	93.72	98.51
car	99.77	100.00	100.00	saheart	99.97	100.00	100.00
chess	99.96	100.00	100.00	satimage	89.55	90.37	92.15
coil2000	93.54	94.50	94.66	segment	97.29	97.40	98.92
contraceptive	53.22	54.92	58.64	sonar	99.85	100.00	100.00
crx	99.62	100.00	100.00	spambase	92.11	92.72	93.80
dermatology	86.84	98.61	100.00	spectfheart	74.93	79.63	85.19
ecoli	96.68	98.51	100.00	splice	98.41	100.00	100.00
flare	99.77	100.00	100.00	texture	94.31	94.55	99.91
german	89.91	100.00	100.00	thyroid	98.60	99.79	100.00
haberman	99.99	100.00	100.00	tic-tac-toe	99.99	100.00	100.00
housevotes	99.72	100.00	100.00	titanic	72.44	80.45	82.27
ionosphere	99.94	100.00	100.00	twonorm	91.11	86.01	98.37
iris	100.00	100.00	100.00	vehicle	99.39	99.41	100.00
letter	69.81	99.72	100.00	vowel	88.92	90.40	97.98
magic	100.00	100.00	100.00	wdbc	99.97	100.00	100.00
mammographic	80.06	81.33	85.54	wine	99.01	100.00	100.00
mushroom	100.00	99.82	100.00	wisconsin	93.34	93.43	97.08
newthyroid	97.88	100.00	100.00	yeast	98.77	99.33	100.00

Table 4
Performance summary for the Krypteia Units.

	Average units	Best unit	Best result
Average Acc.	87.48	94.09	**96.02**

Table 5
Feature importance for the Krypteia ensemble computed using a CART model.

Feature	% of importance
Nº of KNN	7.47%
Nº of LDA	7.61%
Nº of QDA	7.40%
Nº of SVM	8.19%
Nº of trees	5.13%
Unit size	11.55%
VG sample size	7.02%
Survival threshold	39.95%
Full ordeal	2.00%
Survival-only ordeal	1.57%
Social-only ordeal	2.08%

the Krypteia Army to beat the other two by a 1.5% points margin. We have also computed a homogeneous version of the Krypteia army using only SVM classifiers and the arithmetical mean as the aggregation function. This version of the Krypteia performed worst than the rest of the Krypteia configurations, reinforcing the idea that both the heterogeneity and the aggregation functions improve the ensemble performance.

6.7. Results for unit-all and division-all fusion schemes

Instead of performing the decision making scheme of the Krypteia Army, we can fuse all the 300 units using the unit-all or the division-all fusion schemes.

In Fig. 3 we studied the expected performance for the 300 units in the Krypteia Army studied in Section 6.5 using the division-all fusion scheme (fusing n different units as if they were just one division). To do so, we have computed the average accuracy for all the datasets using up to 300 Krypteia Units. These Krypteia Units are the same as those used in previous sections, selected randomly. There we can see that this system seems to reduce performance beyond $n = 10$, and drops significantly when $n > 50$, stabilizing around 92.40%. In no case this scheme performed better than the Krypteia Army.

In Fig. 4 we performed the same experiment using the unit-all (fusing n different subjects as if they were just one unit) decision making phase. A similar situation happens in this case, where a 20–30% of the original army seems to perform greatly when fused in this way, but then stabilizes in an inferior performance. Again, in no case the average accuracy obtained was better than the Krypteia Army.

6.8. Performance for each aggregation function

Each time we use a Krypteia ensemble, we need to choose among one of many aggregation functions. Based on the results from our experiments, we counted the number of times each one of them was the best performing aggregation. By profiling each one the aggregations, we hope to discard the worst ones in future trainings and to reduce the number of possible candidates.

In Fig. 5 we show the times that each of the different aggregations provided the best result in an experiment. We found the classical arithmetic mean to be the one that won most of the times, but with only a very small margin with respect to the maximum, the minimum, the OWA operators, the n-overlap functions, and the Choquet integral. Sugeno integrals performed poorly and the generalizations of the Choquet integral never won in any case. However, it is worth pointing out that the performance of the fuzzy integrals depends on the fuzzy measure used.

7. Comparing the Krypteia ensemble with other ensemble classifiers

In this section we compared the results obtained with the Krypteia ensemble with seven very diverse types of classifier ensembles [86]. Three of them are classical ensemble algorithms: adaboost, bagging, and majority vote using SVMs and random forests. The hyperparameters for these algorithms were fixed according to the parameter values reported in [86]. The other four schemes perform OCS, using oracles, synthetic data, a reference classifier, and meta-features to discard faulty subjects:

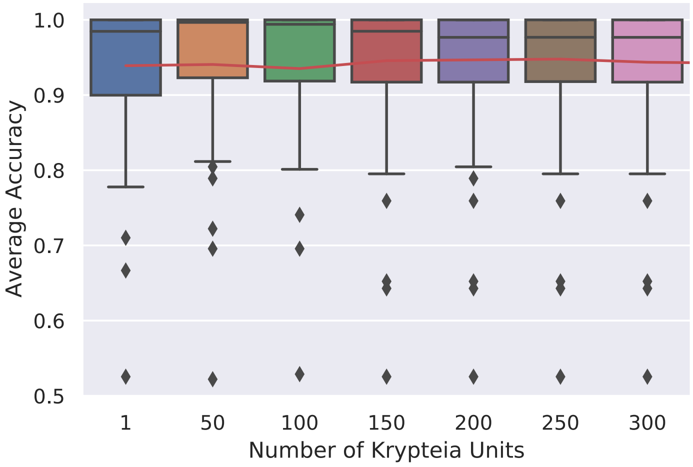


Fig. 3. Average accuracy for all the studied datasets using ensembles of *n* randomly chosen Krypteia Units in the Krypteia Army studied in Section 6.5.

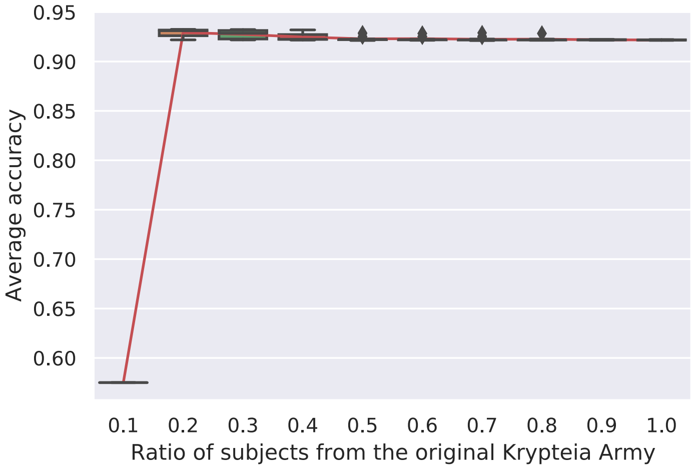


Fig. 4. Average accuracy for all the studied datasets using ensembles of *n* random weak learners from the original Krypteia Army studied in Section 6.5.

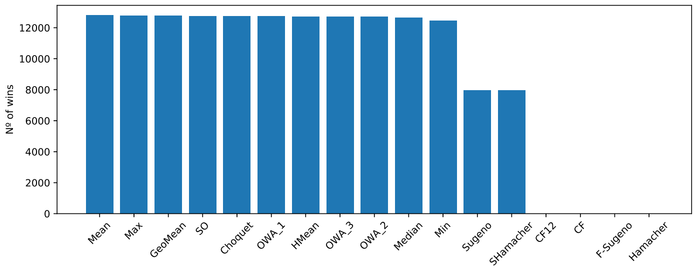


Fig. 5. Number of times each aggregation yielded the best results in all the experiments performed with Krypteia ensembles.

Table 6
Results for the Krypteia divisions.

Dataset	Average	Best division	Best result	Dataset	Average	Best division	Best result
abalone	100.00	100.00	100.00	nursery	89.97	99.96	99.96
appendicitis	85.71	85.71	85.71	optdigits	94.76	95.28	95.37
australian	81.30	81.88	81.88	page-blocks	38.88	97.26	97.26
balance	100.00	100.00	100.00	phoneme	89.64	89.64	89.82
banana	88.61	89.81	89.81	penbased	98.99	99.04	99.40
bands	100.00	100.00	100.00	pima	100.00	100.00	100.00
breast	100.00	100.00	100.00	post-operative	40.00	100.00	100.00
bupa	67.25	66.67	69.57	ring	91.86	90.88	92.57
car	100.00	100.00	100.00	saheart	100.00	100.00	100.00
chess	100.00	100.00	100.00	satimage	91.10	90.83	91.30
coil2000	92.04	92.16	92.47	segment	98.46	98.48	98.70
contraceptive	52.81	52.54	53.56	sonar	100.00	100.00	100.00
crx	100.00	100.00	100.00	spambase	92.74	92.72	93.26
dermatology	86.67	95.83	98.61	spectfheart	75.00	77.78	77.78
ecoli	97.01	97.01	97.01	splice	100.00	100.00	100.00
flare	100.00	100.00	100.00	texture	94.35	94.36	94.91
german	90.00	100.00	100.00	thyroid	99.82	99.86	99.86
haberman	100.00	100.00	100.00	tic-tac-toe	100.00	100.00	100.00
housevotes	100.00	100.00	100.00	titanic	72.41	80.45	80.45
ionosphere	100.00	100.00	100.00	twonorm	92.24	92.57	93.11
iris	100.00	100.00	100.00	vehicle	99.41	99.41	99.41
letter	69.81	99.72	100.00	vowel	90.81	90.91	91.92
magic	100.00	100.00	100.00	wdbc	100.00	100.00	100.00
mammographic	78.98	79.52	80.12	wine	100.00	100.00	100.00
mushroom	100.00	100.00	100.00	wisconsin	93.14	92.70	93.43
newthyroid	97.67	97.67	97.67	yeast	99.73	100.00	100.00

Table 7
Performance summary for the Krypteia divisions.

	Average division	Best division	Best result
Average Acc.	94.09	94.49	**94.58**

- Adaboost [21]: it serially trains each classifier. In each iteration, it weights each instance according to its difficulty to be classified, aiming to correctly classify it in the next iteration. For our experimentation, we have used 50 decision trees to form the Adaboost.
- Bagging (Bootstrap Aggregation) [20]: it aims to increase accuracy by combining the outputs of the classifiers in the ensemble that were trained using different subsamples of the original data. Sampling with replacement is used to train all the classifiers in the ensemble and thus some of the instances may appear more than once in the training set. For our experimentation, we have used 10 decision trees to form the Bagging classifier.
- Majority vote SVM: it consists of different SVM classifiers trained with a different kernel. For our case, we have trained 5 different RBF kernels classifiers with five different σ parameters evenly spaced such as: $[0.5, 1.5]/(number\, of\, features)$.
- Random Forest [87]: it combines the output of many different decision trees computed from different subsamples of the original data. The final decision is taken as the majority vote of all of the trees outputs'. We have set as 100 the number of trees in the Random Forest for our experimentation.
- K-Nearest Oracles Eliminate (K-NORAE) [88]: it selects the classifiers that correctly classify all the samples in their region of competence.
- Dynamic Ensemble Selection Multiclass Imbalance (DES-MI) [89]: it generates artificial training sets of randomly balanced data and then chooses the classifiers that correctly discriminated the minority class samples.
- Randomized Reference Classifier (DES-RRC) [90]: it combines Dynamic Ensemble Selection with a measure to evaluate each possible classifier in the final ensemble using a reference classifier.
- META-DES [53]: it selects a set of classifiers from a list, using five different meta-features to test each classifier's competence.
- Extreme Gradient Boosting (XGBoost) [91]: gradient boosting is a generalization of Adaboost that consist of using a differentiable loss function. This function is optimized using a gradient descent procedure, so that in each step a new weak learner is included to reduce the loss of the system. Gradient boosting is considered to be the state-of-the-art in classification of tabular data [92].

We computed the analogous experiments with these ensemble classifiers as in the Krypteia ensembles. Table 9 collects the average accuracy for all the datasets for the different ensemble classifiers. We found the Krypteia Army to be the most effective classifier, followed by the Best Krypteia Division and the Best Krypteia Unit. All of them outperform the best performing of the seven classifier ensembles tested, DES-MI. In Table 10 we have computed how many times each algorithm performed best (counting ties as winnings for both). We found that most of the times the Krypteia Army showed the best result and that every Krypteia variant again outperform all the benchmarking classifier ensembles.

In Table 11 we show the P-values obtained using a Wilcoxon statistical test, comparing all the ensembles used with the Krypteia Army, which was the ensemble with the highest average accuracy. We found statistical differences favouring the Krypteia Army compared to any of the other classifiers tested.

Finally, we have tested the performance of the Krypteia algorithm in a setting with higher dimensionality and more samples. In order to do so, we have used a dataset that is very popular in the deep learning literature: CIFAR10 [93]. This dataset consists of 600 000 images that belong to 10 different classes, 50 000 used for training and 10 000 for test. In order to apply our proposal to this dataset, we used a LeNet convolutional network architecture [94]. Once the network is trained, we discard the last layer of this network and compute its output for each sample. We use this output as the features to train the different classification algorithms.

In Table 12 we show the accuracy results in the test set for the Krypteia Army and two benchmarking classical ensemble approaches that performed better, Random Forest and Extreme Gradient boosting. We can see that both the Random Forest and the Extreme Gradient Boosting performed worse than the original performance of this LeNet architecture for this dataset, that achieved an accuracy rate of 65.84, while the Krypteia army performed slightly better (65.87) and performed the best overall. This is an interesting result keeping in mind

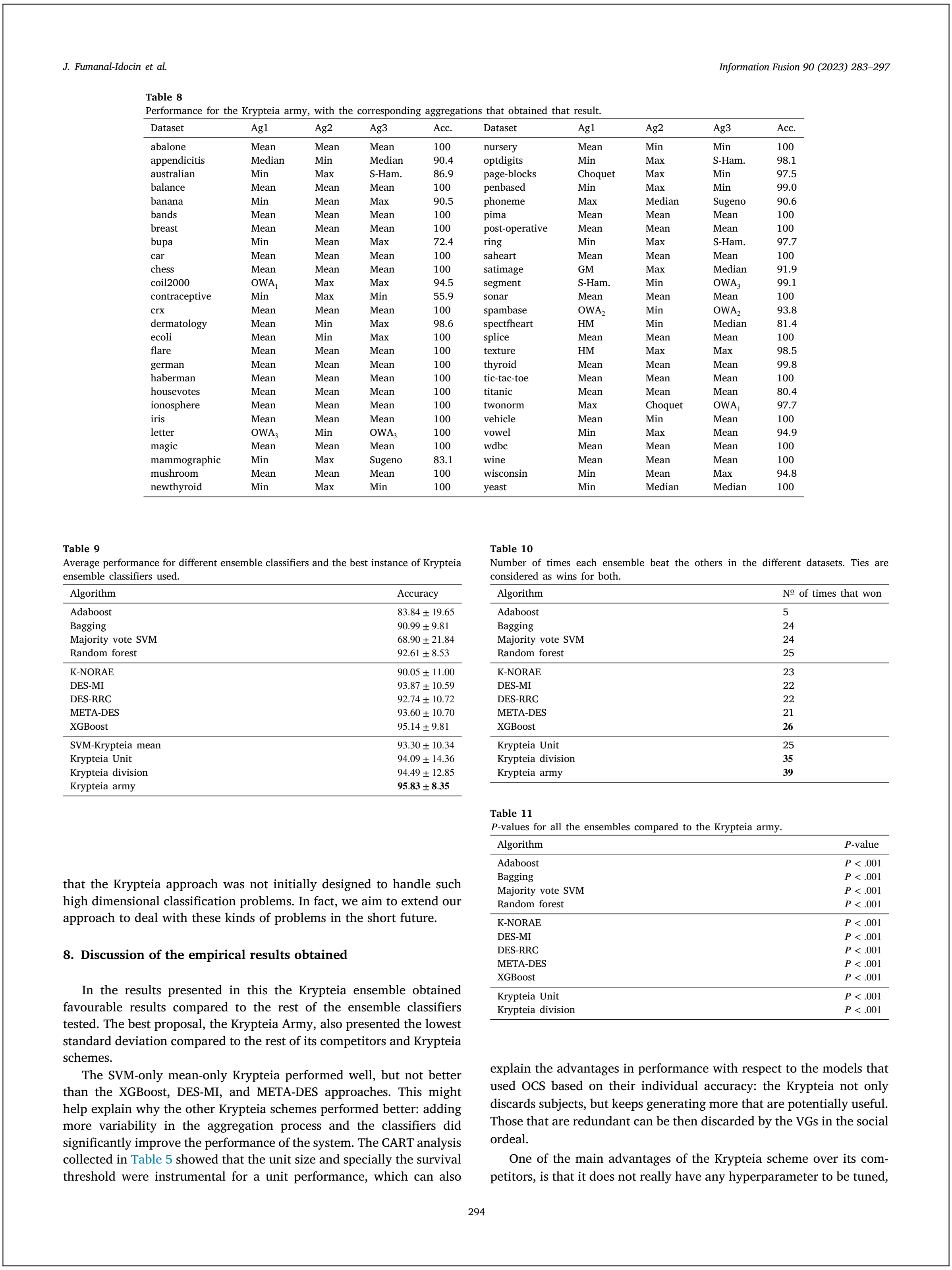


Table 8
Performance for the Krypteia army, with the corresponding aggregations that obtained that result.

Dataset	Ag1	Ag2	Ag3	Acc.	Dataset	Ag1	Ag2	Ag3	Acc.
abalone	Mean	Mean	Mean	100	nursery	Mean	Min	Min	100
appendicitis	Median	Min	Median	90.4	optdigits	Min	Max	S-Ham.	98.1
australian	Min	Max	S-Ham.	86.9	page-blocks	Choquet	Max	Min	97.5
balance	Mean	Mean	Mean	100	penbased	Min	Max	Min	99.0
banana	Min	Mean	Max	90.5	phoneme	Max	Median	Sugeno	90.6
bands	Mean	Mean	Mean	100	pima	Mean	Mean	Mean	100
breast	Mean	Mean	Mean	100	post-operative	Mean	Mean	Mean	100
bupa	Min	Mean	Max	72.4	ring	Min	Max	S-Ham.	97.7
car	Mean	Mean	Mean	100	saheart	Mean	Mean	Mean	100
chess	Mean	Mean	Mean	100	satimage	GM	Max	Median	91.9
coil2000	OWA$_1$	Max	Max	94.5	segment	S-Ham.	Min	OWA$_3$	99.1
contraceptive	Min	Max	Min	55.9	sonar	Mean	Mean	Mean	100
crx	Mean	Mean	Mean	100	spambase	OWA$_2$	Min	OWA$_2$	93.8
dermatology	Mean	Min	Max	98.6	spectfheart	HM	Min	Median	81.4
ecoli	Mean	Min	Max	100	splice	Mean	Mean	Mean	100
flare	Mean	Mean	Mean	100	texture	HM	Max	Max	98.5
german	Mean	Mean	Mean	100	thyroid	Mean	Mean	Mean	99.8
haberman	Mean	Mean	Mean	100	tic-tac-toe	Mean	Mean	Mean	100
housevotes	Mean	Mean	Mean	100	titanic	Mean	Mean	Mean	80.4
ionosphere	Mean	Mean	Mean	100	twonorm	Max	Choquet	OWA$_1$	97.7
iris	Mean	Mean	Mean	100	vehicle	Mean	Min	Mean	100
letter	OWA$_3$	Min	OWA$_3$	100	vowel	Min	Max	Mean	94.9
magic	Mean	Mean	Mean	100	wdbc	Mean	Mean	Mean	100
mammographic	Min	Max	Sugeno	83.1	wine	Mean	Mean	Mean	100
mushroom	Mean	Mean	Mean	100	wisconsin	Min	Mean	Max	94.8
newthyroid	Min	Max	Min	100	yeast	Min	Median	Median	100

Table 9
Average performance for different ensemble classifiers and the best instance of Krypteia ensemble classifiers used.

Algorithm	Accuracy
Adaboost	83.84 ± 19.65
Bagging	90.99 ± 9.81
Majority vote SVM	68.90 ± 21.84
Random forest	92.61 ± 8.53
K-NORAE	90.05 ± 11.00
DES-MI	93.87 ± 10.59
DES-RRC	92.74 ± 10.72
META-DES	93.60 ± 10.70
XGBoost	95.14 ± 9.81
SVM-Krypteia mean	93.30 ± 10.34
Krypteia Unit	94.09 ± 14.36
Krypteia division	94.49 ± 12.85
Krypteia army	**95.83 ± 8.35**

Table 10
Number of times each ensemble beat the others in the different datasets. Ties are considered as wins for both.

Algorithm	Nº of times that won
Adaboost	5
Bagging	24
Majority vote SVM	24
Random forest	25
K-NORAE	23
DES-MI	22
DES-RRC	22
META-DES	21
XGBoost	**26**
Krypteia Unit	25
Krypteia division	**35**
Krypteia army	**39**

Table 11
P-values for all the ensembles compared to the Krypteia army.

Algorithm	*P*-value
Adaboost	$P < .001$
Bagging	$P < .001$
Majority vote SVM	$P < .001$
Random forest	$P < .001$
K-NORAE	$P < .001$
DES-MI	$P < .001$
DES-RRC	$P < .001$
META-DES	$P < .001$
XGBoost	$P < .001$
Krypteia Unit	$P < .001$
Krypteia division	$P < .001$

that the Krypteia approach was not initially designed to handle such high dimensional classification problems. In fact, we aim to extend our approach to deal with these kinds of problems in the short future.

8. Discussion of the empirical results obtained

In the results presented in this the Krypteia ensemble obtained favourable results compared to the rest of the ensemble classifiers tested. The best proposal, the Krypteia Army, also presented the lowest standard deviation compared to the rest of its competitors and Krypteia schemes.

The SVM-only mean-only Krypteia performed well, but not better than the XGBoost, DES-MI, and META-DES approaches. This might help explain why the other Krypteia schemes performed better: adding more variability in the aggregation process and the classifiers did significantly improve the performance of the system. The CART analysis collected in Table 5 showed that the unit size and specially the survival threshold were instrumental for a unit performance, which can also explain the advantages in performance with respect to the models that used OCS based on their individual accuracy: the Krypteia not only discards subjects, but keeps generating more that are potentially useful. Those that are redundant can be then discarded by the VGs in the social ordeal.

One of the main advantages of the Krypteia scheme over its competitors, is that it does not really have any hyperparameter to be tuned,

Table 12
Accuracy values for the CIFAR10 dataset. A LeNet model is considered to compute the features for each image.

Algorithm	Accuracy
LeNet base performance	65.84
Extreme Gradient boosting	64.09
Random forest	65.37
Krypteia army	65.87

as almost every parameter is sampled from a random distribution. The only parameters that cannot be stochastically fixed are the number of units in the Krypteia Army and the aggregation functions used, that we chose using a five-fold validation on the training set. On the contrary, it has quite different steps, which makes its computation more complex than its rivals. The CART analysis performed in this study could be a good starting point to choose those elements that could be simplified without affecting performance.

Although there is not a theoretical limit for the size of the problems the Krypteia scheme can tackle, this scheme is designed for data that can fit completely in memory. When using other paradigms for datasets of bigger sizes, we must take into account further consideration to fully support and exploit mini-batches or map-reduce approaches. For example, it is possible to use classifiers that perform some kind of online-learning, or it is also possible to use dedicated classifiers for each batch. Besides, the use of VGs can be focused in keeping the classifiers that performed best on the data that is significantly different to what the system has already processed. The performance of all these options needs to be tested properly in order to determine which is the best option to scale the Krypteia scheme to large databases and will be a subject of study in future works.

9. Conclusions and future lines

In this paper we presented the Krypteia ensemble: a new form to generate classifier ensembles based on an ancient Spartan ritual to train the future elites of their society. We detailed the process needed to compute these classifiers and we explained how they relate to this ancient tradition, by exposing each different subject to distinct hardships. Then, we studied how different aggregation functions work in different Krypteia ensembles.

We tested the different Krypteia ensembles on a large experimental study including 52 datasets. We have studied the performance of different forms of the Krypteia ensemble, and the effect of the various parameters that define the Krypteia training process. Then, we compared the results obtained by the Krypteia against 7 other ensemble design algorithms, obtaining significantly better results.

Future research shall aim to improve the way in which Krypteia Units are assembled to form Krypteia Divisions. We are also interested in expanding the Krypteia scheme to support environments where the data size is too big to fit in memory, like in deep learning, where the dataset is loaded in mini-batches, and big data. Refining the performance of the different ordeals, i.e. adding label perturbation and different loss functions to the survival ordeal, will also be explored. Finally, we also intend to study the application of the Krypteia and other social phenomena to other domains different from classification [95].

CRediT authorship contribution statement

J. Fumanal-Idocin: Conceptualization, Methodology, Software, Writing – original draft. **O. Cordón:** Writing – review & editing. **H. Bustince:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

Javier Fumanal Idocin and Humberto Bustince's research has been supported by project PID2019-108392 GB I00 (AEI/10.13039/501100011033).

Oscar Cordón's research has been funded by the Spanish Ministry of Science and Innovation (MICIN), Agencia Estatal de Investigación (AEI), Spain, under grant CONFIA (PID2021-122916NB-I00), and by the Regional Government of Andalusia under grant EXAISFI (P18-FR-4262), both including European Regional Development Funds (ERDF).

Open access funding provided by Universidad Pública de Navarra.

References

[1] M. Berthold, D.J. Hand, Intelligent Data Analysis, Springer, 2003.
[2] S. Brandt, Data Analysis, Springer, 1998.
[3] D.A. Keim, Information visualization and visual data mining, IEEE Trans. Vis. Comput. Graphics 8 (1) (2002) 1–8.
[4] R. Cooley, B. Mobasher, J. Srivastava, Web mining: Information and pattern discovery on the world wide web, in: Proceedings Ninth IEEE International Conference on Tools with Artificial Intelligence, IEEE, 1997, pp. 558–567.
[5] N. Zhong, Y. Li, S.-T. Wu, Effective pattern discovery for text mining, IEEE Trans. Knowl. Data Eng. 24 (1) (2010) 30–44.
[6] J.-P. Brunet, P. Tamayo, T.R. Golub, J.P. Mesirov, Metagenes and molecular pattern discovery using matrix factorization, Proc. Natl. Acad. Sci. 101 (12) (2004) 4164–4169.
[7] M.M. Petrou, C. Petrou, Image Processing: The Fundamentals, John Wiley & Sons, 2010.
[8] C.C. Aggarwal, Data classification, in: Data Mining, Springer, 2015, pp. 285–344.
[9] H. Ramchoun, M.A.J. Idrissi, Y. Ghanou, M. Ettaouil, Multilayer perceptron: Architecture optimization and training, Int. J. Interact. Multimedia Artif. Intell. 4 (1) (2016) 26–30.
[10] A.F.M. Hani, H.A. Nugroho, H. Nugroho, Gaussian Bayes classifier for medical diagnosis and grading: application to diabetic retinopathy, in: 2010 IEEE EMBS Conference on Biomedical Engineering and Sciences, IECBES, IEEE, 2010, pp. 52–56.
[11] D. Ververidis, C. Kotropoulos, Emotional speech classification using Gaussian mixture models, in: 2005 IEEE International Symposium on Circuits and Systems, IEEE, 2005, pp. 2871–2874.
[12] L.E. Peterson, K-nearest neighbor, Scholarpedia 4 (2) (2009) 1883.
[13] C. Cortes, V. Vapnik, Support vector machine, Mach. Learn. 20 (3) (1995) 273–297.
[14] N. Saleena, et al., An ensemble classification system for twitter sentiment analysis, Procedia Comput. Sci. 132 (2018) 937–946.
[15] X. Feng, Z. Xiao, B. Zhong, J. Qiu, Y. Dong, Dynamic ensemble classification for credit scoring using soft probability, Appl. Soft Comput. 65 (2018) 139–151.
[16] C.B.C. Latha, S.C. Jeeva, Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques, Inform. Med. Unlocked 16 (2019) 100203.
[17] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1–2) (2010) 1–39.
[18] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45.
[19] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[20] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[21] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[22] A. Rojarath, W. Songpan, C. Pong-inwong, Improved ensemble learning for classification techniques based on majority voting, in: 2016 7th IEEE International Conference on Software Engineering and Service Science, ICSESS, IEEE, 2016, pp. 107–110.
[23] R.M. Cruz, D.V. Oliveira, G.D. Cavalcanti, R. Sabourin, FIRE-DES++: Enhanced online pruning of base classifiers for dynamic ensemble selection, Pattern Recognit. 85 (2019) 149–160.
[24] A. Onan, S. Korukoğlu, H. Bulut, A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification, Expert Syst. Appl. 62 (2016) 1–16.

[25] J.M. Moyano, E.L. Gibaja, K.J. Cios, S. Ventura, Review of ensembles of multi-label classifiers: models, experimental study and prospects, Inf. Fusion 44 (2018) 33–45.

[26] D. Partridge, W.B. Yates, Engineering multiversion neural-net systems, Neural Comput. 8 (4) (1996) 869–893.

[27] L. Wang, T. Mo, X. Wang, W. Chen, Q. He, X. Li, S. Zhang, R. Yang, J. Wu, X. Gu, et al., A hierarchical fusion framework to integrate homogeneous and heterogeneous classifiers for medical decision-making, Knowl.-Based Syst. 212 (2021) 106517.

[28] G. Beliakov, H. Bustince, T. Calvo, A Practical Guide to Averaging Functions, Vol. 329, Springer, 2016.

[29] G.P. Dimuro, J. Fernández, B. Bedregal, R. Mesiar, J.A. Sanz, G. Lucca, H. Bustince, The state-of-art of the generalizations of the Choquet integral: from aggregation and pre-aggregation to ordered directionally monotone functions, Inf. Fusion 57 (2020) 27–43.

[30] S. Abbaszadeh, E. Hullermeier, Machine learning with the Sugeno Integral: The case of binary classification, IEEE Trans. Fuzzy Syst. (2020).

[31] L. De Miguel, M. Sesma-Sara, M. Elkano, M. Asiain, H. Bustince, An algorithm for group decision making using n-dimensional fuzzy sets, admissible orders and OWA operators, Inf. Fusion 37 (2017) 126–131.

[32] L. De Miguel, D. Gómez, J.T. Rodríguez, J. Montero, H. Bustince, G.P. Dimuro, J.A. Sanz, General overlap functions, Fuzzy Sets and Systems 372 (2019) 81–96.

[33] H. Bustince, G. Beliakov, G.P. Dimuro, B. Bedregal, R. Mesiar, On the definition of penalty functions in data aggregation, Fuzzy Sets and Systems 323 (2017) 1–18.

[34] S.T. Hadjitodorov, L.I. Kuncheva, L.P. Todorova, Moderate diversity for better cluster ensembles, Inf. Fusion 7 (3) (2006) 264–275.

[35] V. Bolón-Canedo, A. Alonso-Betanzos, Ensembles for feature selection: A review and future trends, Inf. Fusion 52 (2019) 1–12.

[36] A.J. Bayliss, The Spartans, Oxford University Press, 2020.

[37] W. Wang, Some fundamental issues in ensemble methods, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 2243–2250.

[38] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Mach. Learn. 51 (2) (2003) 181–207.

[39] Q. Dai, R. Ye, Z. Liu, Considering diversity and accuracy simultaneously for ensemble pruning, Appl. Soft Comput. 58 (2017) 75–91.

[40] L. Liu, W. Wei, K.-H. Chow, M. Loper, E. Gursoy, S. Truex, Y. Wu, Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness, in: 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems, MASS, IEEE, 2019, pp. 274–282.

[41] A. Rahman, S. Tasnim, Ensemble classifiers and their applications: A review, 2014, arXiv preprint arXiv:1404.4088.

[42] H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, V. Vapnik, Boosting and other ensemble methods, Neural Comput. 6 (6) (1994) 1289–1301.

[43] L. Rokach, O. Maimon, I. Lavi, Space decomposition in data mining: A clustering approach, in: International Symposium on Methodologies for Intelligent Systems, Springer, 2003, pp. 24–31.

[44] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo, A. Alonso-Betanzos, Ensemble feature selection: homogeneous and heterogeneous approaches, Knowl.-Based Syst. 118 (2017) 124–139.

[45] Z.H. Kilimci, S. Akyokus, Deep learning-and word embedding-based heterogeneous classifier ensembles for text classification, Complexity 2018 (2018).

[46] J.N. van Rijn, G. Holmes, B. Pfahringer, J. Vanschoren, The online performance estimation framework: heterogeneous ensemble learning for data streams, Mach. Learn. 107 (1) (2018) 149–176.

[47] R. Lysiak, M. Kurzynski, T. Woloszynski, Optimal selection of ensemble classifiers using measures of competence and diversity of base classifiers, Neurocomputing 126 (2014) 29–35, Recent trends in Intelligent Data Analysis Online Data Processing.

[48] H.G. Zefrehi, H. Altınçay, Imbalance learning using heterogeneous ensembles, Expert Syst. Appl. 142 (2020) 113005.

[49] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, Dynamic classifier selection: Recent advances and perspectives, Inf. Fusion 41 (2018) 195–216.

[50] V.Y. Kulkarni, P.K. Sinha, Pruning of random forest classifiers: A survey and future directions, in: 2012 International Conference on Data Science & Engineering, ICDSE, IEEE, 2012, pp. 64–68.

[51] R. Hu, S. Zhou, Y. Liu, Z. Tang, Margin-based Pareto ensemble pruning: An ensemble pruning algorithm that learns to search optimized ensembles, Comput. Intell. Neurosci. 2019 (2019).

[52] C. Lin, W. Chen, C. Qiu, Y. Wu, S. Krishnan, Q. Zou, LibD3C: ensemble classifiers with a clustering and dynamic selection strategy, Neurocomputing 123 (2014) 424–435.

[53] R.M. Cruz, R. Sabourin, G.D. Cavalcanti, T.I. Ren, META-DES: A dynamic ensemble selection framework using meta-learning, Pattern Recognit. 48 (5) (2015) 1925–1935.

[54] K. Trawiński, O. Cordón, A. Quirin, A study on the use of multiobjective genetic algorithms for classifier selection in FURIA-based fuzzy multiclassifiers, Int. J. Comput. Intell. Syst. 5 (2) (2012) 231–253.

[55] K. Trawinski, O. Cordón, A. Quirin, Embedding evolutionary multiobjective optimization into fuzzy linguistic combination method for fuzzy rule-based classifier ensembles, in: 2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, 2014, pp. 1968–1975.

[56] R.C.T. de Souza, C.A. de Macedo, L. dos Santos Coelho, J. Pierezan, V.C. Mariani, Binary coyote optimization algorithm for feature selection, Pattern Recognit. 107 (2020) 107470.

[57] M. Elkano, M. Galar, J.A. Sanz, P.F. Schiavo, S. Pereira Jr., G.P. Dimuro, E.N. Borges, H. Bustince, Consensus via penalty functions for decision making in ensembles in fuzzy rule-based classification systems, Appl. Soft Comput. 67 (2018) 728–740.

[58] L. Rokach, Pattern Classification using Ensemble Methods, Vol. 75, World Scientific, 2010.

[59] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets, Inform. Sci. 354 (2016) 178–196.

[60] J. Fumanal-Idocin, Y.-K. Wang, C.-T. Lin, J. Fernández, J.A. Sanz, H. Bustince, Motor-imagery-based brain-computer interface using signal derivation and aggregation functions, IEEE Trans. Cybern. (2021) 1–12, http://dx.doi.org/10.1109/TCYB.2021.3073210.

[61] G. Lucca, J.A. Sanz, G.P. Dimuro, B. Bedregal, H. Bustince, R. Mesiar, CF-integrals: A new family of pre-aggregation functions with application to fuzzy rule-based classification systems, Inform. Sci. 435 (2018) 94–110.

[62] F. Bardozzo, B. De La Osa, L. Horanská, J. Fumanal-Idocin, M. delli Priscoli, L. Troiano, R. Tagliaferri, J. Fernandez, H. Bustince, Sugeno integral generalization applied to improve adaptive image binarization, Inf. Fusion 68 (2021) 37–45.

[63] G. Beliakov, J.-Z. Wu, Learning fuzzy measures from data: simplifications and optimisation strategies, Inform. Sci. 494 (2019) 100–113.

[64] K. Trawiński, O. Cordón, L. Sánchez, A. Quirin, A genetic fuzzy linguistic combination method for fuzzy rule-based multiclassifiers, IEEE Trans. Fuzzy Syst. 21 (5) (2013) 950–965.

[65] M. Papčo, I. Rodríguez-Martínez, J. Fumanal-Idocin, A.H. Altalhi, H. Bustince, A fusion method for multi-valued data, Inf. Fusion 71 (2021) 1–10.

[66] M. Gupta, J. Qi, Theory of T-norms and fuzzy inference methods, Fuzzy Sets and Systems 40 (3) (1991) 431–450.

[67] R.R. Yager, J. Kacprzyk, The Ordered Weighted Averaging Operators: Theory and Applications, Springer Science & Business Media, 2012.

[68] G.P. Dimuro, G. Lucca, B. Bedregal, R. Mesiar, J.A. Sanz, C.-T. Lin, H. Bustince, Generalized cf1f2-integrals: From Choquet-like aggregation to ordered directionally monotone functions, Fuzzy Sets and Systems 378 (2020) 44–67.

[69] H. Bustince, R. Mesiar, J. Fernandez, M. Galar, D. Paternain, A. Altalhi, G. Dimuro, B. Bedregal, Z. Takáč, D-Choquet integrals: Choquet integrals based on dissimilarities, Fuzzy Sets and Systems 414 (2021) 1–27, Aggregation Functions.

[70] G. Lucca, J. Antonio Sanz, G. Pereira Dimuro, B. Bedregal, R. Mesiar, A. Kolesarova, H. Bustince, Preaggregation functions: Construction and an application, IEEE Trans. Fuzzy Syst. 24 (2) (2016) 260–272.

[71] M. Sugeno, Theory of Fuzzy Integrals and its Applications (Ph.D. thesis), Tokyo Institute of Technology, 1974.

[72] L.-W. Ko, Y.-C. Lu, H. Bustince, Y.-C. Chang, Y. Chang, J. Ferandez, Y.-K. Wang, J.A. Sanz, G.P. Dimuro, C.-T. Lin, Multimodal fuzzy fusion for enhancing the motor-imagery-based brain computer interface, IEEE Comput. Intell. Mag. 14 (1) (2019) 96–106.

[73] D. Futter, Plutarch, Plato and Sparta, Akroterion 57 (1) (2012) 35–51.

[74] J. Ducat, Spartan Education: Youth and Society in the Classical Period, ISD LLC, 2006.

[75] I. Joseph, The deadliest blogger: Military history page, 2021, https://deadliestblogpage.wordpress.com/. Accessed: 2021-04-26.

[76] A.J. Izenman, Linear discriminant analysis, in: Modern Multivariate Statistical Techniques, Springer, 2013, pp. 237–280.

[77] S. Srivastava, M.R. Gupta, B.A. Frigyik, Bayesian quadratic discriminant analysis, J. Mach. Learn. Res. 8 (Jun) (2007) 1277–1305.

[78] Y. Freund, L. Mason, The alternating decision tree learning algorithm, in: International Conference on Machine Learning, Vol. 99, 1999, pp. 124–133.

[79] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.

[80] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 20–29.

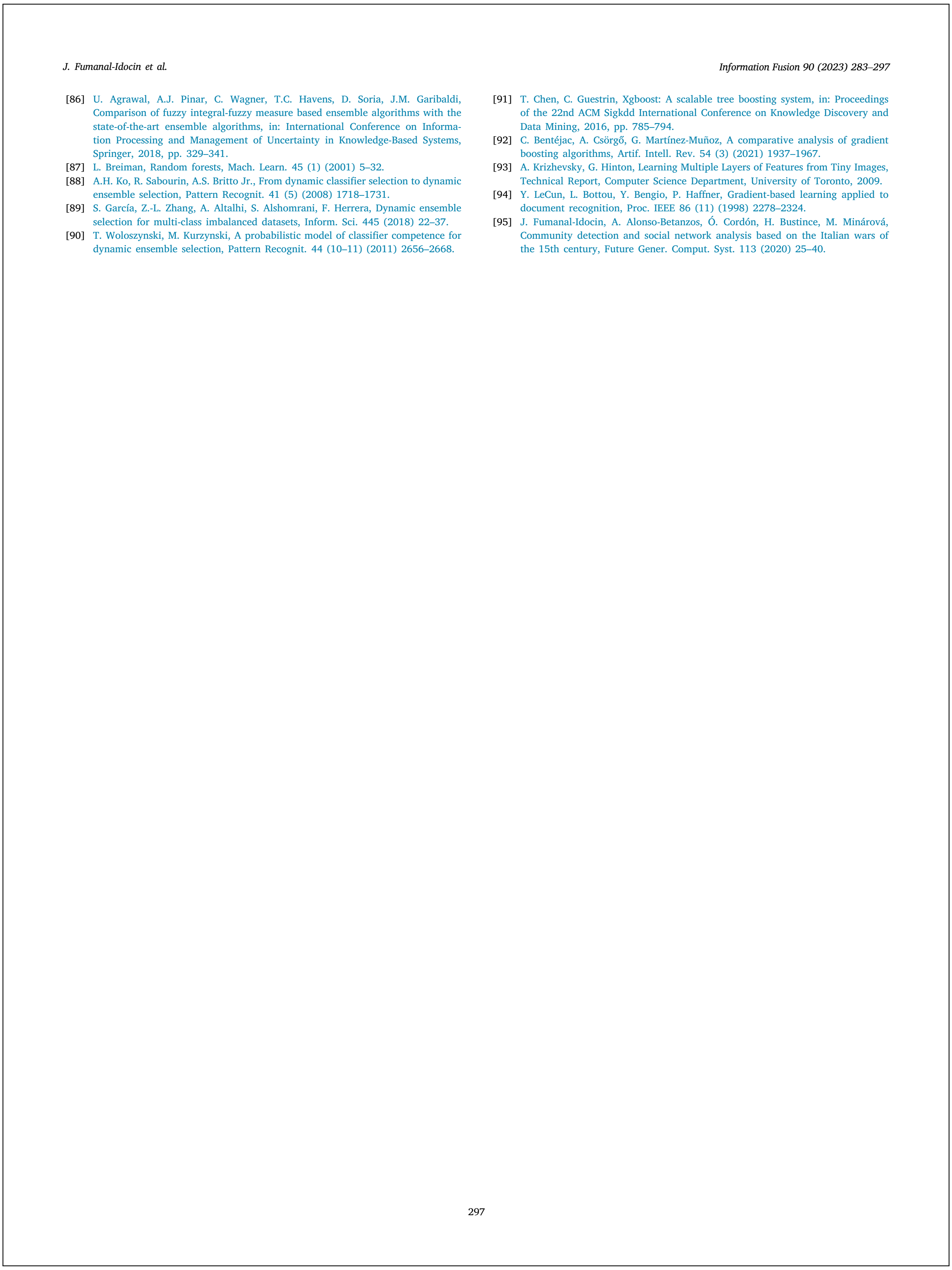
[81] D.L. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Trans. Syst. Man Cybern. SMC-2 (3) (1972) 408–421.

[82] X.-S. Yang, S. Deb, S. Fong, Metaheuristic algorithms: optimal balance of intensification and diversification, Appl. Math. Inf. Sci. 8 (3) (2014) 977.

[83] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Mult.-Valued Logic Soft Comput. 17 (2011).

[84] E. Bradford, Thermopylae: The Battle for the West, Open Road Media, 2014.

[85] E.A. Antipov, E.B. Pokryshevskaya, Mass appraisal of residential apartments: An application of random forest for valuation and a CART-based approach for model diagnostics, Expert Syst. Appl. 39 (2) (2012) 1772–1778.



[86] U. Agrawal, A.J. Pinar, C. Wagner, T.C. Havens, D. Soria, J.M. Garibaldi, Comparison of fuzzy integral-fuzzy measure based ensemble algorithms with the state-of-the-art ensemble algorithms, in: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer, 2018, pp. 329–341.

[87] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[88] A.H. Ko, R. Sabourin, A.S. Britto Jr., From dynamic classifier selection to dynamic ensemble selection, Pattern Recognit. 41 (5) (2008) 1718–1731.

[89] S. García, Z.-L. Zhang, A. Altalhi, S. Alshomrani, F. Herrera, Dynamic ensemble selection for multi-class imbalanced datasets, Inform. Sci. 445 (2018) 22–37.

[90] T. Woloszynski, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, Pattern Recognit. 44 (10–11) (2011) 2656–2668.

[91] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd ACM Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[92] C. Bentéjac, A. Csörgő, G. Martínez-Muñoz, A comparative analysis of gradient boosting algorithms, Artif. Intell. Rev. 54 (3) (2021) 1937–1967.

[93] A. Krizhevsky, G. Hinton, Learning Multiple Layers of Features from Tiny Images, Technical Report, Computer Science Department, University of Toronto, 2009.

[94] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

[95] J. Fumanal-Idocin, A. Alonso-Betanzos, Ó. Cordón, H. Bustince, M. Minárová, Community detection and social network analysis based on the Italian wars of the 15th century, Future Gener. Comput. Syst. 113 (2020) 25–40.

6.11 ARTxAI: Explainable Artificial Intelligence Curates Deep Representation Learning of Artistic Images

Associated publication:

- Fumanal-Idocin, J., Andreu-Perez J. Hagras H., Cordón O., & Bustince, H. (2022). ARTxAI: Explainable Artificial Intelligence Curates Deep Representation Learning of Artistic Images *Information Fusion.*

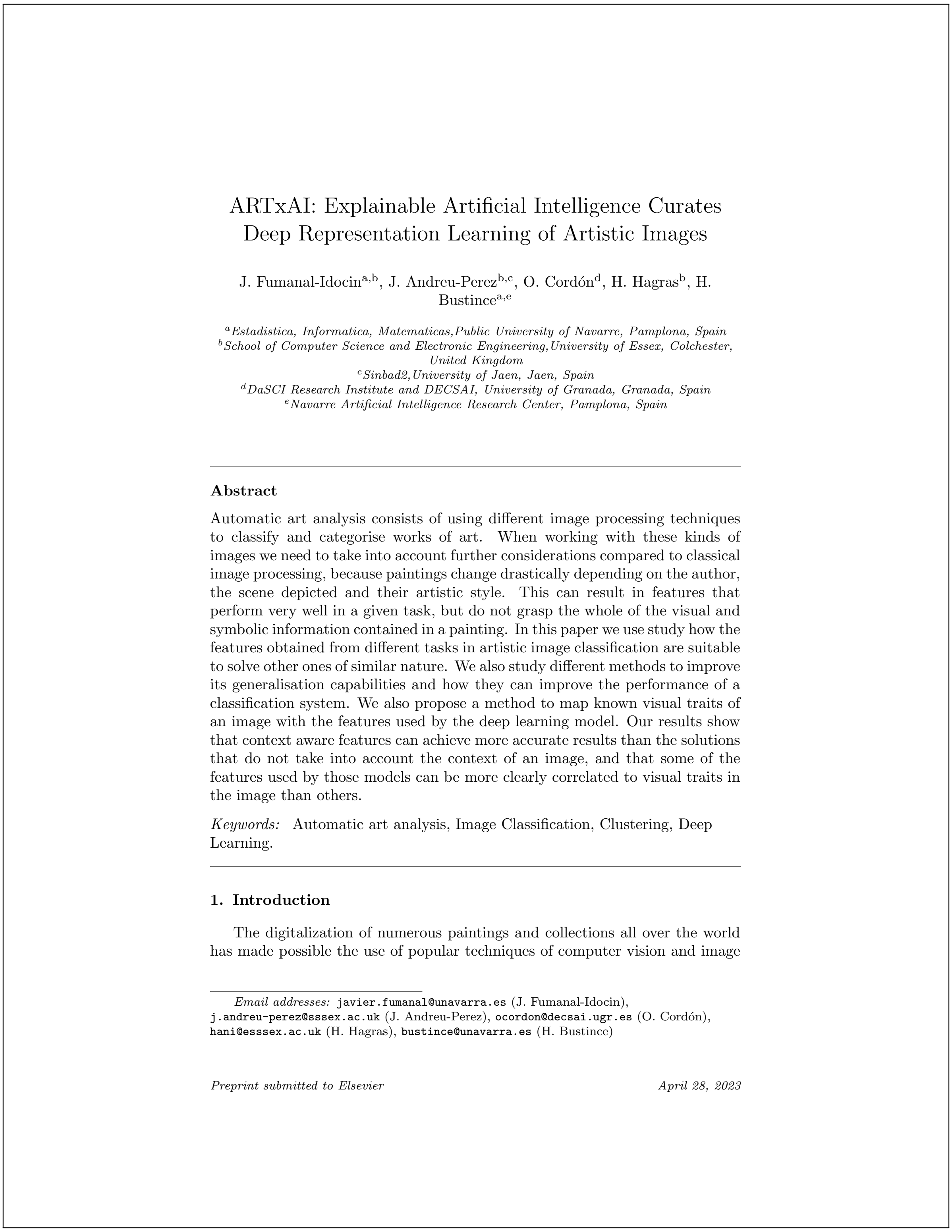
  Status: Submitted.

  Impact Factor (JCR 2021): 17.564.

  Categories:

  Computer Science, Artificial Intelligence. Ranking 4/144 (Q1).

  Computer Science, Theory & Methods. Ranking 1/109 (Q1).



ARTxAI: Explainable Artificial Intelligence Curates Deep Representation Learning of Artistic Images

J. Fumanal-Idocin[a,b], J. Andreu-Perez[b,c], O. Cordón[d], H. Hagras[b], H. Bustince[a,e]

[a] *Estadistica, Informatica, Matematicas,Public University of Navarre, Pamplona, Spain*
[b] *School of Computer Science and Electronic Engineering,University of Essex, Colchester, United Kingdom*
[c] *Sinbad2,University of Jaen, Jaen, Spain*
[d] *DaSCI Research Institute and DECSAI, University of Granada, Granada, Spain*
[e] *Navarre Artificial Intelligence Research Center, Pamplona, Spain*

Abstract

Automatic art analysis consists of using different image processing techniques to classify and categorise works of art. When working with these kinds of images we need to take into account further considerations compared to classical image processing, because paintings change drastically depending on the author, the scene depicted and their artistic style. This can result in features that perform very well in a given task, but do not grasp the whole of the visual and symbolic information contained in a painting. In this paper we use study how the features obtained from different tasks in artistic image classification are suitable to solve other ones of similar nature. We also study different methods to improve its generalisation capabilities and how they can improve the performance of a classification system. We also propose a method to map known visual traits of an image with the features used by the deep learning model. Our results show that context aware features can achieve more accurate results than the solutions that do not take into account the context of an image, and that some of the features used by those models can be more clearly correlated to visual traits in the image than others.

Keywords: Automatic art analysis, Image Classification, Clustering, Deep Learning.

1. Introduction

The digitalization of numerous paintings and collections all over the world has made possible the use of popular techniques of computer vision and image

Email addresses: `javier.fumanal@unavarra.es` (J. Fumanal-Idocin), `j.andreu-perez@sssex.ac.uk` (J. Andreu-Perez), `ocordon@decsai.ugr.es` (O. Cordón), `hani@esssex.ac.uk` (H. Hagras), `bustince@unavarra.es` (H. Bustince)

processing on artistic data [4]. One of the most promising topics in this direction is the automatic analysis of paintings, in which these techniques are applied in creative tasks historically performed on most of the galleries and museums. Some of these are author verification [20], style analysis [39] and restoration [66].

Artistic image processing was traditionally performed using hand-crafted or ad-hoc features [11]. However, the advent of deep learning and convolutional neural networks has made automatically extracted features very popular [34, 58, 21]. Usually, these models are pre-trained and then fine-tuned for each specific task [3, 2, 50]. This is specially important for the case of artistic images [14].One of the actual limitations of these models is that human experts perform their analysis based not only on the visual cues, but they also rely on their knowledge on the historical context, other paintings, materials, etc. [40]. The addition of contextual and historical information to visual cues has been studied to perform different classification tasks in artistic image analysis [29, 27, 62]. However, there is not a standard procedure to extract the contextual information associated to each artistic work. Besides, sometimes context is not encoded in well defined labels. When the information is not well structured, like in a textual commentary, it is also necessary to discriminate those parts relevant to the task.

One of the most popular approaches to encode this kind of information are knowledge graphs [29, 27, 62]. A knowledge graph captures the relationships between different concepts and attributions using the structure of a network [57, 16]. Indeed, graphs are a popular form of representing information [43] and they have been used to solve a myriad of problems in different areas of knowledge, like computer science [44, 7, 23], biology [45] and the social sciences [8, 25]. However, when using a knowledge graph, a continuous space representation must be constructed from the nodes in the graph. This process is usually performed using deep learning models like node2vec [33, 32]. Another possibility consists of using multi-task learning, in which a set of different related tasks are trained together so that the information obtained from one is also used in the others [61].

Capturing a painting context is also useful to improve the features obtained with a CNN. ResNet50 has proven to have good generalisation capabilities when trained on the extensively used Imagenet dataset. This generalisation capability is specially important in tasks where there is an significant domain shift, and when visual information must be interpreted correctly in order to detect abstract concepts in the image [31]. The focus of the current paper is to study how general are the features used in an artistic image classification problem, and how useful they are when applied in other similar tasks (i.e. how useful they are to develop transfer learning). Doing so, we also measure if the network is learning "shortcuts" to solve the task instead of finding significant patterns, which can constitute a significant performance metric of deep learning models [60]. We also want to test if the features of obtained from a black box model can be correlated to known characteristics in the original image.

In order to achieve these aims, we study different ways to obtain such features, using only visual cues of the image and when additional information is also available. We also propose a new way to represent the contextual embeddings from different paintings using fuzzy memberships that expands previous approaches in this sense [28, 26]. We shall study how the Fuzzy C-Means clustering algorithm [6] and an adapted version of a fuzzy-rule based fuzzy clustering algorithm can be used to construct an embedding space, and how this embedding captures relevant information from the original texts. We shall also study the use of Contrastive Language-Image Pre-Training (CLIP) features [17, 49]. In order to map the deep features obtained with these models to known visual cues, we will use approximate reasoning through the means of fuzzy rules.

The rest of the paper is organized as follows: in Section 2 we recall some of the previous concepts required to understand this work and review some relevant literature. In Section 3 we introduce the framework for artistic image classification using contextual embeddings and the different methods proposed to obtain contextual embeddings from textual annotations. Then, in Section 4 we design the experimental setup and discuss the results obtained using the different contextual-aware and non-contextual aware-methods. Subsequently, in Section 5 we describe our method to explain deep features. Finally, in Section 6 we give some final conclusions and future lines for this work.

2. Background

In this section we review some previous works regarding the Fuzzy C-Means clustering algorithm, clustering using fuzzy rules, representation learning and artistic artwork classification.

2.1. Fuzzy C-Means

Fuzzy C-Means (FCM) is a well known fuzzy clustering algorithm, in which each element is assigned not only to one group, but rather presents a fuzzy membership to each of the groups considered [18].

FCM aims to minimise the following objective function:

$$\arg\min \sum_{i=1}^{n} \sum_{j=1}^{c} w_{ij} \|\mathbf{x}_i - \mathbf{c}_j\|^2 \tag{1}$$

where n is the number of observations, c is the number of different clusters, m is a constant, c_j is a cluster, and $\mathbf{x}_i$ is an observation. Finally, w_{ij} is the membership of the $i - th$ observation to the $j - th$ cluster, that follows this expression:

$$w_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{||\mathbf{x}_i - \mathbf{c}_j||}{||\mathbf{x}_i - \mathbf{c}_k||} \right)^2}. \tag{2}$$

The algorithm assigns randomly a coefficient for each observation to each cluster. Then, it computes the centroid for each cluster, and computes each membership again. The process is repeated until convergence. There is a need to provide a cluster number c as input to the method.

2.2. Fuzzy rule-based classification and clustering

Fuzzy rule-based classification consists of discriminate observations into different categories using rules that follow this structure [56]:

$$\text{IF } \mathbf{x}_1 \text{ is } \mathbf{a}_{j1} \ldots \mathbf{x}_n \text{ is } \mathbf{a}_{jn} \text{ THEN class } j \text{ for } j = 1, \ldots, c \tag{3}$$

where $\mathbf{x}$ is a multidimensional vector, j is the consequent class, $\mathbf{a}_j$ is an antecedent linguistic value for class j, and c is the number of different classes. For this purpose, each attribute is reescalated into the $[0, 1]$ unit interval, and then, the n different attributes are partitioned into different fuzzy subpartitions.

There are different ways in which these fuzzy subpartitions can be generated [19]. Given them, there are also different algorithms to generate a set of fuzzy rules to classify the samples [52]. It is also possible to use fuzzy rules to perform clustering [42].

2.3. Representation learning

Representation learning consists of automatically extracting and computing features suitable for machine learning tasks from unstructured data like text, videos, or images [5]. Deep learning is one of the most popular fields in which feature learning is performed. Convolutional neural networks have been massively popular tools to embed images and video into vector spaces [35] as well as text [46].

Just as images, videos and text, networks can also be embedded into vector spaces using deep learning models [33, 53, 64]. Deep learning models can be combined with other classical methods in text processing, using term frequency-inverse document frequency (TF-IDF) indexing or Latent Dirichlet Allocation (LDA) [9, 38]. Deep feature extraction has been very succesful in classical machine learning approaches, but some problems still remain, like the the lack of a clear geometrical intuition in the latent space. This problem was partially solved using Variational autoencoders [47], which instead of learning directly to encode the samples, they learn the data *a priori* distribution. Unfortunately, these approaches still present some problems [65].

2.4. Artistic artwork classification

Historically, automatic art analysis has been performed using handcrafted features that relied on colour, brushwork, and scale-invariant features [11, 36]. Then, once the features were extracted they were used to train different kinds of classifiers. The most popular tasks include identifying the author, the style, and the theme of a painting [28].

The advent of deep learning has substituted the use of manual features by automatically extracted ones [12]. These features have been extracted using different networks, like the Residual Network (ResNet) [59] and the VGG16 [55]. Pre-trained networks can be used to recognize different shapes and entities in images, and they have been extensively used in art analysis as well. It is also possible to fine-tune these networks to those tasks [12, 28].

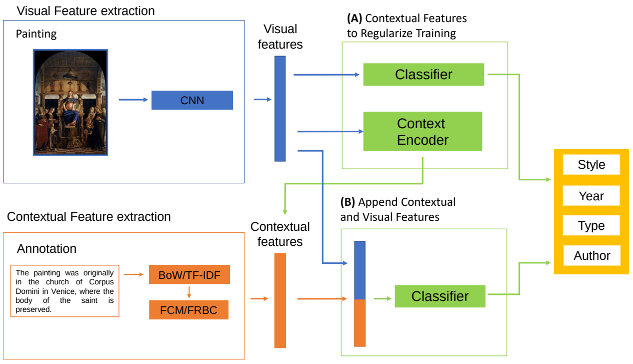

Visual Feature extraction
Painting
CNN
Visual features
(A) Contextual Features to Regularize Training
Classifier
Context Encoder
Style
Year
Type
Author
Contextual Feature extraction
Annotation
The painting was originally in the church of Corpus Domini in Venice, where the body of the saint is preserved.
BoW/TF-IDF
FCM/FRBC
Contextual features
(B) Append Contextual and Visual Features
Classifier

Figure 1: Scheme of the two proposed classification frameworks using contextual features. Option (**a**) uses the contextual features to regularise the training of the CNN. The last layer of the network is replaced to fit the number of classes in the task. Option (**b**) appends the visual and contextual features in a single vector that is fed to a final classification layer.

To further study art from a semantic perspective, visual information can be combined with contextual information from the art pieces. There are different ways in which this information can be incorporated into the classification framework. One possibility is to train simultaneously different classification tasks in a multi-task setting [28]. In this way, features learnt to classify one image can help learning in other tasks and *vice versa*. It is also possible to use knowledge graphs constructed from the dataset itself [27] or from external information [13].

Although these successes, some of these methods still present some shortcomings. It is difficult to tell when the predictions are based on meaningful artistic knowledge of shortcuts. Besides, the categorisation of specific parts of an image (like a cat, a dog, etc.) is different from aspects like an author, which cannot be necessarily correlated only to specific parts of the image. This makes some explainability models like Grad-CAM [54] less useful for this task.

3. Artistic Image Classification Framework

Our proposed framework consists of two different parts (see Figure 1). On the one hand, we use a ResNet 50 [59] to extract visual features for each image. The last layer of the ResNet is substituted by a linear classifier with the appropriate number of classes as the output.

On the other hand, we compute the contextual embeddings using one of the methods proposed:

1. Node2vec in a knowledge graph that connects the paintings according to their shared attributes [28].

2. Fuzzy memberships over the contextual annotations for each painting (More details about this in Section 3.1).

3. Features obtained with a CLIP autoencoder on the contextual annotations.

All of these methods result in a contextual vector representation for each painting. In order to combine the visual and contextual features, we opted for two different approaches:

- We use the contextual information vector in order to "regularise" the visual features. To do so, we have two "final" layers: one encoder that transforms the final feature vector of the network into the contextual features, and another one that performs the classification. These encoders are single full connected layers with a Rectified Linear Unit activation function (Figure 1a).

- We append the contextual information vector to the visual characteristics vector. Then, we use a full connected layer to learn from the resulting vector (Figure 1b).

The loss for each class c is the standard cross-entropy:

$$l_c(y, \hat{y}) = -\frac{1}{n} \sum_{i=1}^{n} y_{ci} \log \hat{y}_{ci} + (1 - y_{ci}) \log(1 - \hat{y}_{ci}) \tag{4}$$

Given r, the final embedding obtained from the ResNet, and m the number of clusters obtained with the fuzzy clustering, the loss function for the reconstruction of the fuzzy memberships vector is the Smooth L1:

$$\delta_{emb}(a, b) = \begin{cases} \frac{1}{2}(a - b), & \text{if } |a - b| \leq 1 \\ |a - b| - \frac{1}{2}, & \text{otherwise} \end{cases} \tag{5}$$

$$l_{emb} = \sum_{i=1}^{n} \sum_{j=1}^{m} \delta_{emb}(w_{ij}, r_{ij}) \tag{6}$$

3.1. Fuzzy context encoding

The idea of using fuzzy clustering for this task is that the space formed using a word embedding method can be a faithful representation of the original domain, but it might not be useful to solve the task at hand. Since we are interested in using these features to discriminate between classes, we are more interested in the topology of the representation obtained and the groups that are naturally present in them.

We expect that these groups agglomerate categories that are not mutually exclusive. For example, in the case of artistic representation, style and year can be very correlated because of artistic movements. There are many more possible examples in this case: e.g. landscapes can be grouped together but belong to different authors, etc. Fuzzy clustering is the most suitable clustering tool for this task, since we intend to express the membership to different, not mutually exclusive groups. For each observation, we have a fuzzy membership degree for each of the pertinent groups.

Besides, fuzzy clustering is much more convenient than the traditional K-means algorithm for this task since fuzzy memberships are all in the same $[0, 1]$ range. This means that the resulting vector will be a real-valued vector, with more information than a one-hot encoding corresponding to a unique cluster. Besides, memberships to clusters that are far away can be modelled with a 0 in both cases, while distances in the K-means can be very different in magnitude but represent the same thing: the observation is not in this cluster. Thus, fuzzy memberships are also more suitable than Euclidean distances in a hard clustering approach for this task.

In order to compute the fuzzy memberships, we first encode the text annotations using BoW or TF-IDF encodings.Once this codification is computed, we run a fuzzy clustering algorithm to obtained the desired number of memberships.

The most popular fuzzy clustering algorithm is the FCM [6], which consist of an adaption of the popular K-Means that obtains a fuzzy clustering configuration. Fuzzy rules have also been intensively used to perform classification and data mining. However, according to our knowledge, no algorithm to obtain a fuzzy clustering configuration using rules exists. In order to use a fuzzy rule-based algorithm to compute fuzzy clustering, we have adapted the algorithm proposed by Mansoori et al. in [42]. This algorithm results in a crisp clustering configuration, so we changed it in order to obtain the desired fuzzy memberships. More details about this can be found in Appendix B.

4. Classification results

In this section we evaluate the performance of the different proposed method in the artistic image classification problem, using the visual embeddings from the ResNet and the context aware embeddings. We also compare which solution offers the best result in its original application. We also measure how effective these features are when solving other similar but different tasks.

We have first evaluated the accuracy for the four different tasks (type, school, timeframe and author identification) classification methods:

1. The ResNet50 and the VGG16 networks using their correspondent pre-trained weights. We adapt the last layer to match the number of target classes. These solutions only consider the visual information for each image.

2. The ResNet50 and the VGG16 fine-tuned in a multi-task setting (MTL) setting for all the different classes, so that context is captured by the

shared information between tasks. We retrained all the weights for each network.

3. The ResNet50 with information captured from contextual annotations and metadata, using node2vec representations represented by a knowledge graph (KGM) [27].

4. Our proposed classification framework in Figure 1 using the ResNet50 to extract visual features and BoW/TF-IDF and FCM to encode the textual annotations. We use the contextual features as a regularising element in the training process, and appending both vectors of features (marked as "append" in Table 1). For the case of the BoW codification, we also test a lighter model that uses only the top 100 most popular words.

5. Our proposed classification framework in Figure 1 using the ResNet50 to extract visual features and BoW/TF-IDF and FRBC to encode the textual annotations. We use the contextual features as a regularising element in the training process, and appending both vectors of features (marked as "append" in Table 1).

6. Our proposed classification framework in Figure 1 using the ResNet50 to extract visual features and a CLIP autoencoder to encode the textual annotations.

4.1. Datasets

For our experimentation we have used the SemArt dataset [29]. This dataset consists of 21,384 painting images. Following the original data partition in [29], 19,244 images are used for training (i.e. a 90%), $1,069$ for validation, and $1,069$ for test (i.e. a 5% each). Each painting has associated a textual artistic comment. In this dataset four different classification tasks are proposed:

- Type: each painting is classified according to 10 different common types of paintings: portrait, landscape, religious, etc.

- School: each painting is identified with different schools of art: Italian, Dutch, French, Spanish, etc. There are a total of 25 classes of this kind.

- Timeframe: The attribute Timeframe, which corresponds to periods of 50 years evenly distributed between 801 and 1900, is used to classify each painting according to its creation date. We consider only the timeframes where at least 10 paintings are present. This corresponds to 18 classes.

- Author: corresponds to the author of each paintings. We consider a total of 350 painters, that comprise the set of authors with more than 10 paintings in the dataset.

A in-depth analysis of the contextual annotations of the SemArt dataset can be found in Appendix A.

We have also used the WikiArt dataset. This dataset is a collection of high-resolution images of artworks, along with their associated metadata, that were scraped from Wikipedia [51].

The WikiArt dataset contains over 81,000 images of fine-art paintings, representing a wide range of artistic styles and historical periods, from the 11th century to the present day. Each image in the dataset is accompanied by a set of metadata, including the title of the artwork, the artist, the year of creation, the medium used and the dimensions of the artwork, among other attributes.

4.2. Results for the classification tasks

Table 1 shows the results each of the tasks and models. We can see that the MTL methods performed worst than those that used a KGM or FCM to capture contextual information. Comparing the KGM and FCM encodings, the BoW-FCM performed better in all tasks but “Author”. When taking into account contextual information using MTL, KGM and FCM-based methods, the performance improved substantially for all classes.

The best result for each different task was obtained using an FCM-based model. However, those that used most words for the contextual embedding performed poorly on the Author task, in which the BoW model with only 100 words performed significantly better than the rest of the FCM-based models. It was also the best performing method for this class compared to the KGM and MTL-based proposals. This could be due to the fact that the “Author” classification is the most complicated task, with only a few learning examples per class, and the available contextual vectors are not specific enough to help discriminate in those cases. Appending the contextual vector instead of using it to regularise the gradient seems to have a similar effect in the final performance of the system. Since we are not guaranteed to have textual annotations, it is preferable to use those that only required them in the training process.

Finally, we have also joined both paradigms using a MTL model with the two different contextual vectors as a regularising element. This model outperformed the rest of the models, as they could access more information than the rest.

4.3. Results for reusing deep features with a SVM

In order to study how general are features obtained in these frameworks we measure how useful they are to solve the other tasks by extracting the features for each image and then traning a SVM classifier for the rest of the tasks studied. For example, to test how good the features obtained for the author task using the BoW+FCM model are for the type task, we extract the features obtained with this model for the former task, and then we trained a SVM that uses these features to solve the latter. We do this for two of the context-aware models. We also test a new approach which consists of using the features obtained using a CLIP autoencoder for both the text annotation and the image.

Table 2 shows the results of training each SVM model for each pair of possible source and target class. In this case the deep features vector obtained consists of 20 different features and performing feature selection. Results in this case

Table 1: Correct Classification Ratio results for the different attributes on SemArt Dataset.

Method	Type	School	TimeFrame	Author
VGG16	0.706	0.502	0.418	0.482
ResNet50	0.726	0.557	0.456	0.500
VGG16 MTL	0.732	0.585	0.497	0.513
ResNet50 MTL	0.763	0.565	0.464	0.431
ContextNet	0.786	0.647	0.597	0.548
BoW + FCM	0.794	0.655	0.604	0.238
BoW + FCM-apppend	0.802	0.654	0.584	0.230
TF-IDF + FCM	0.786	0.645	0.604	0.229
TF-IDF + FCM-append	0.778	0.654	0.589	0.226
BoW_{100} + FCM	0.792	0.630	0.586	**0.559**
TF-IDF + FRBC	0.785	0.643	0.597	0.233
TF-IDF + FRBC-append	0.759	0.623	0.533	0.154
CLIP-context	0.784	0.649	0.601	0.215
MTL-FCM	**0.804**	**0.691**	0.618	0.531
MTL-CLIP	0.790	0.677	**0.630**	0.551

show similar results for any pair of source and target tasks. Results in training set differs considerably in some cases. For example, for the Author class in some cases we can obtain up to 0.540 of accuracy in training set, which evidences an important overfit in the training set. The CLIP-only features did not perform as good as the deep features from the Resnet except in the Timeframe task. However, they did not overfit so much as training and test performance is much more similar. This means that from the general dataset that trained the CLIP model it is impossible to solve these specific tasks. On the other hand, the deep features generated specifically form a space too complex to exploit using the geometrical intuitions of the SVM [47].

5. Explaining Deep Features

In previous section we discussed the importance of the different features used for different tasks, and which ones were more general and relied less on shortcuts. However, the interpretation of such features is still unknown, as they are the output from a large number of matrix operations. It is possible to grasp a better understanding of their behaviour if we correlate them to known characteristics in the original image.

In order to do so, we propose to use a Fuzzy Rule Based Classifier (FRBC) that will map known features to the degree of activation of the deep features used to classify the paintings in each task. This will allow us to understand

Table 2: Results for a SVM trained on the features computed originally for another task using feature selection over vectors of originally 20 features. Each row shows the original task and each column the target one.

Framework	Original task	Type	School	TimeFrame	Author
BoW+FCM	Type	0.167	0.382	0.001	0.152
	School	0.167	0.160	0.001	0.369
	TimeFrame	0.031	0.385	0.001	0.369
	Author	0.174	0.392	0.009	0.378
CLIP-context	Type	0.149	0.166	0.000	0.383
	School	0.366	0.167	0.000	0.381
	TimeFrame	0.366	0.053	0.009	0.385
	Author	0.369	0.168	0.000	0.384
CLIP-SVM	-	0.227	0.268	0.120	0.247

the predictions done by the network using abstract concepts and Grad-CAM heatmaps [54].

5.1. Extracting style info for SemArt paintings

To extract known features we first use another ResNet50 model trained to recognise artistic movements in a painting. The output of this model will be a vector containing the score for all possible styles considered. This model is trained on the WikiArt dataset, on 100,000 paintings belonging to 2,300 different authors, and that contains 27 different artistic styles. We trained the ResNet for a maximum of 300 epochs with a limit of 50 iterations without a tangible improve in the model performance. It obtained an overall 53% accuracy overall in the Wikiart dataset.

Once we have trained this model, we use it to characterise the sytle of the SemArt paintings. We do not expect a significant domain shift between both datasets as the SemArt dataset was not collected with a particular bias in the selection process. However, in order to check how good the performance was in this dataset, we compared the results using the painters that have paintings in both datasets.

We computed the percentage of times that a painting was classified in Semart with a style in which the author did not have any in Wikiart (Table 3). We found a total 8 common artists in both datasets, with different degrees of missclasification. Of course, some errors are more important than others i. e. it is not the same to incorrectly classify a puntillism painting as impressionist rather than puntillist. However, the whole complexity of this problem is left open in this work.

We found the results satisfactory, as only one painter found significant missclasification: Albert Durer. The rest of the presented an error of less than 0.10 or very few paintings (Singer Sargent).

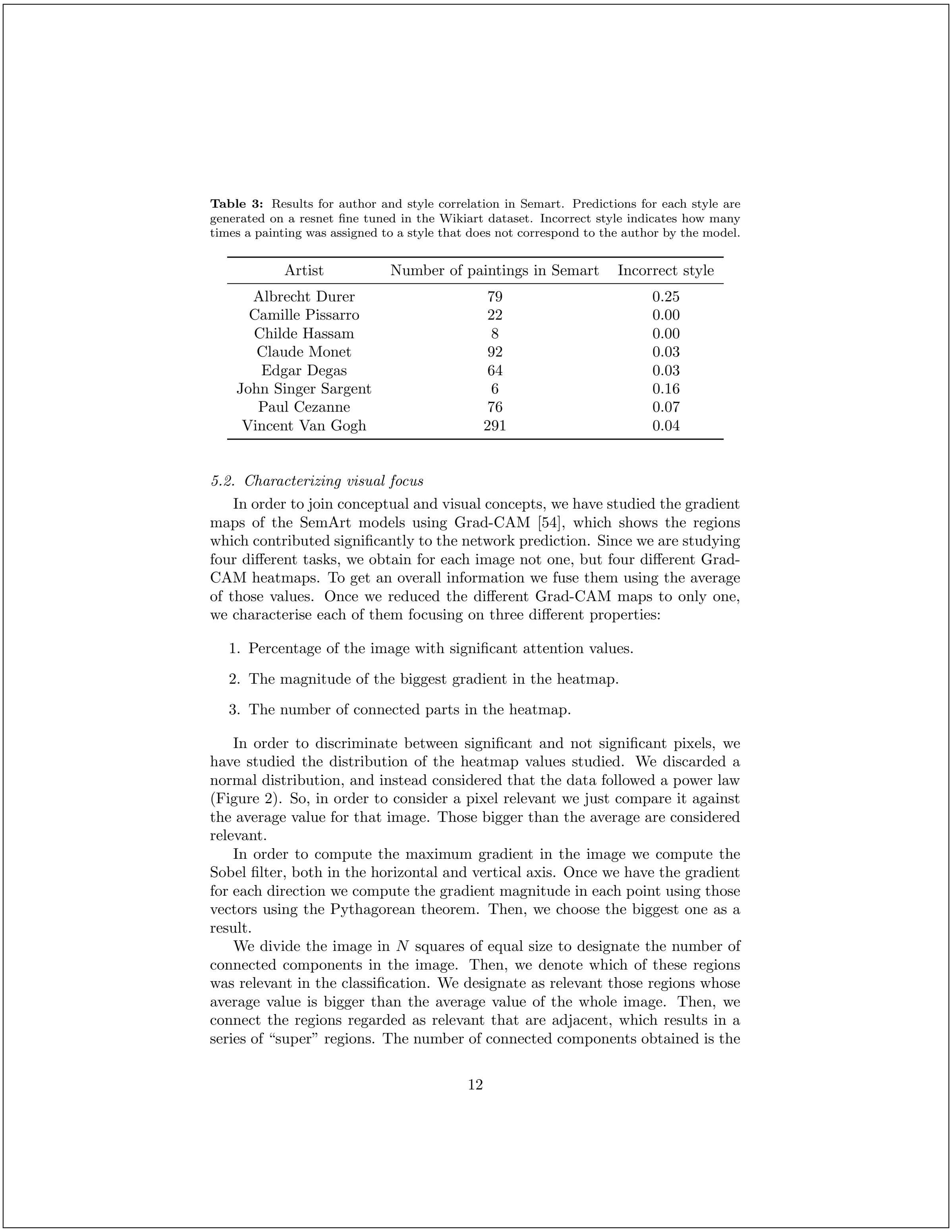


Table 3: Results for author and style correlation in Semart. Predictions for each style are generated on a resnet fine tuned in the Wikiart dataset. Incorrect style indicates how many times a painting was assigned to a style that does not correspond to the author by the model.

Artist	Number of paintings in Semart	Incorrect style
Albrecht Durer	79	0.25
Camille Pissarro	22	0.00
Childe Hassam	8	0.00
Claude Monet	92	0.03
Edgar Degas	64	0.03
John Singer Sargent	6	0.16
Paul Cezanne	76	0.07
Vincent Van Gogh	291	0.04

5.2. Characterizing visual focus

In order to join conceptual and visual concepts, we have studied the gradient maps of the SemArt models using Grad-CAM [54], which shows the regions which contributed significantly to the network prediction. Since we are studying four different tasks, we obtain for each image not one, but four different Grad-CAM heatmaps. To get an overall information we fuse them using the average of those values. Once we reduced the different Grad-CAM maps to only one, we characterise each of them focusing on three different properties:

1. Percentage of the image with significant attention values.
2. The magnitude of the biggest gradient in the heatmap.
3. The number of connected parts in the heatmap.

In order to discriminate between significant and not significant pixels, we have studied the distribution of the heatmap values studied. We discarded a normal distribution, and instead considered that the data followed a power law (Figure 2). So, in order to consider a pixel relevant we just compare it against the average value for that image. Those bigger than the average are considered relevant.

In order to compute the maximum gradient in the image we compute the Sobel filter, both in the horizontal and vertical axis. Once we have the gradient for each direction we compute the gradient magnitude in each point using those vectors using the Pythagorean theorem. Then, we choose the biggest one as a result.

We divide the image in N squares of equal size to designate the number of connected components in the image. Then, we denote which of these regions was relevant in the classification. We designate as relevant those regions whose average value is bigger than the average value of the whole image. Then, we connect the regions regarded as relevant that are adjacent, which results in a series of “super” regions. The number of connected components obtained is the

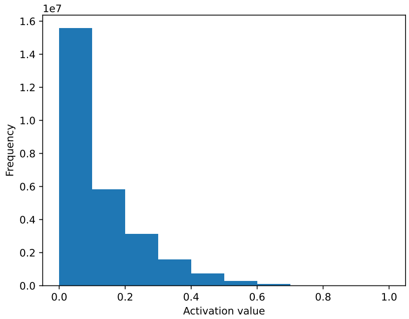


Figure 2: Histogram of the heatmap intensities obtained using Grad-CAM for the SemArt dataset.

Table 4: Statistics of the Grad-CAM heatmap descriptors for the SemArt datase.

Descriptor	Average value	Standard deviation
Maximum gradient	2.27	0.49
Relevant area	0.37	0.08
Number of super regions	2.90	1.39

same as the number of the "super" regions formed in the image. Figure 3 shows an example of the results for this characterisation for one image.

Finally, Table 4 shows a summary of the statistics of these descriptors. We can see that the average value of the relevant parts of the image is about a third, and that there is an average of 3 connected components in each painting.

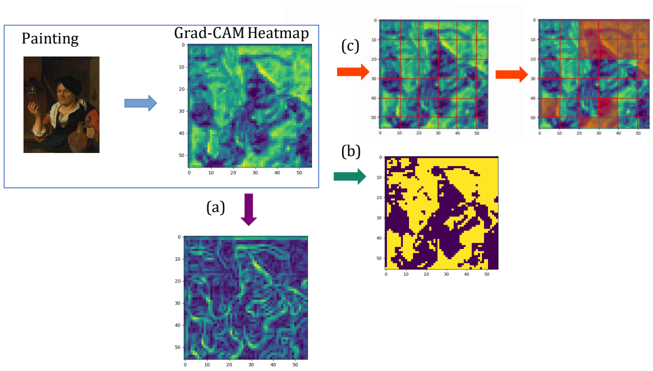
5.3. Mapping known features to deep features

Here, we discuss how we constructed the FRBC and the results obtained in the SemArt dataset discriminating deep features and authors.

5.3.1. Fuzzy rule based classifier used

In order to design the FRBC, we set as 30 the maximum number of rules, 4 the maximum number of antecedents. The linguistic labels are three: low, medium and high. To compute the prediction for a sample, we need to compute the dominance score of each rule r [37]. The dominance score for each rule is the product of their support:

$$s_r(Ants_r \to Cons_r) = \frac{\sum_{\mathbf{x}\in(Ants_r \to Cons_r)} w_r(\mathbf{x})}{|R|}, \tag{7}$$



Descriptor	Value
Maximum gradient	1.8
Relevant area	0.53
Number of super regions	3

Figure 3: Example of the extraction of the different descriptors for a Grad-CAM heatmap. (a) Shows the gradient magnitude for each pixel. We choose the highest value from that image as the descriptor. (b) Shows the pixels denoted as relevant because their value was higher than the average value in the image. (c) Shows the division into different regions and those designed as relevant. Then, we can generate the “super” regions. In the adjacent table, we can find the numeric values for each descriptor in this image.

where $w_p(x)$ is the degree of truth of the antecedents of rule r for the sample x and R is the set of all rules in the FRBC. We are using Mamdani inference, so the firing strength of the rule is the product of the truth degrees of all antecedents of the rule. Confidence is defined as:

$$c_r(Ants_r \rightarrow Cons_r) = \frac{\sum_{\mathbf{x} \in (Ants_r \rightarrow Cons_r)} w_r(\mathbf{x})}{\sum_{r=1, \mathbf{x} \in (Ants_r)} w_r(\mathbf{x})} \tag{8}$$

.

So, the dominance score (DS) of each rule, ds_r, is defined as:

$$ds_r = s_r * c_r \tag{9}$$

Finally, we compute the association degree, $as_r(x)$, using $w_r(\mathbf{x})$ and ds_r:

$$as_r(x) = w_r(\mathbf{x}) * ds_r. \tag{10}$$

Each sample is classified according to the consequent class of the rule with the highest association degree for that sample:

$$P(x) = Cons_{arg\,max(as_r(x) \forall r \in R)} \tag{11}$$

For our experimentation, we have focused on the performance of a standard fuzzy rules based classifier. In order to train one, we used a genetic algorithm that optimises the fuzzy partitions and the antecedents and consequent for each rule. The metric to optimise is the Matthew correlation coefficient:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{12}$$

where TP is true positive, TN means true negative, FP is false positive and FN is false negative.

We also add another condition: in case two subjects performed equally on the fitness metric, we prefer those that did so using less rules.

5.3.2. Mapping known characteristics to deep features

Our aim is to obtain rules that map known characteristics to a relevant activation in a deep feature, like this:

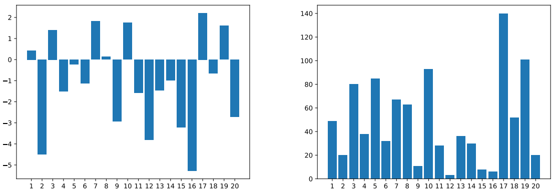
Antecedents	Consequent (Dominant feature)
IF Expressionism IS High AND Relevant area IS Small	Deep Feature 1

Our first step to construct such rules is to designate which features are for each sample, and how useful are in the classification task. In order to so, we have computed the average value and the number of times each feature presented the biggest value in a sample (Figure 4). Some features are clearly more dominant than others i.e. 17 and 19, while others are remarkably low i.e. 12. Although it would be possible to designate more than one feature as dominant for one sample, we have simplified this task as chose as dominant for one sample the deep feature with biggest activation. We have checked the feasibility of this task by visualizing using PCA for each deep feature studied (Figure 5).

Then, we used a Gradient boosting classifier (GBC) [15] to solve the classification task resulted in a 0.22 accuracy. Since Gradient boosting is considered state-of-the-art in the standard tabular classification [24] we can interpret this 0.22 as a upper bound of accuracy in a FRBC. However, we can convert this problem using a One-versus-All scheme. In this way, instead of one multi-class problem we have 20 binary classification problems. As these problems are heavily imbalanced, we use the MCC to evaluate the results for each feature. These are shown in Table 5.

The lack of positive samples for each feature in comparison with the negative ones affects deeply the performance of the GBC. In order to give a more reliable estimation of the performance of the system we subsampled randomly a balanced partition for each feature (i.e. we perform random oversampling, Table 5, column 2). Using these models, we checked the importance that they gave to each style for their predictions. Based on the relevant ones (when the importance value is bigger than the average), we use a FRBC that learns the correspondent rules to map from the input data to the desired class for each deep feature.

Using this model and fuzzy linguistic variables, we can characterise each painting according to their expressiveness of their visual traits. Table 6 shows the MCC for each of the features classification using a FRBC. As expected from Figure 8 visualisations, the performance is very different among the features. Finally, in Table 7 we show the resulting obtained for some of the features were the classification was most successful and in Figure 6 we show an example image for each dominant feature studied where those rules fired.



(a)

(b)

Figure 4: Study of deep features activations. (a)Average value of the 20 deep features used in MTL-FCM predictions.(b) Histogram containing the number of times that each feature presented the biggest value for each sample in the train set.

5.3.3. Mapping known features to discriminate particular authors

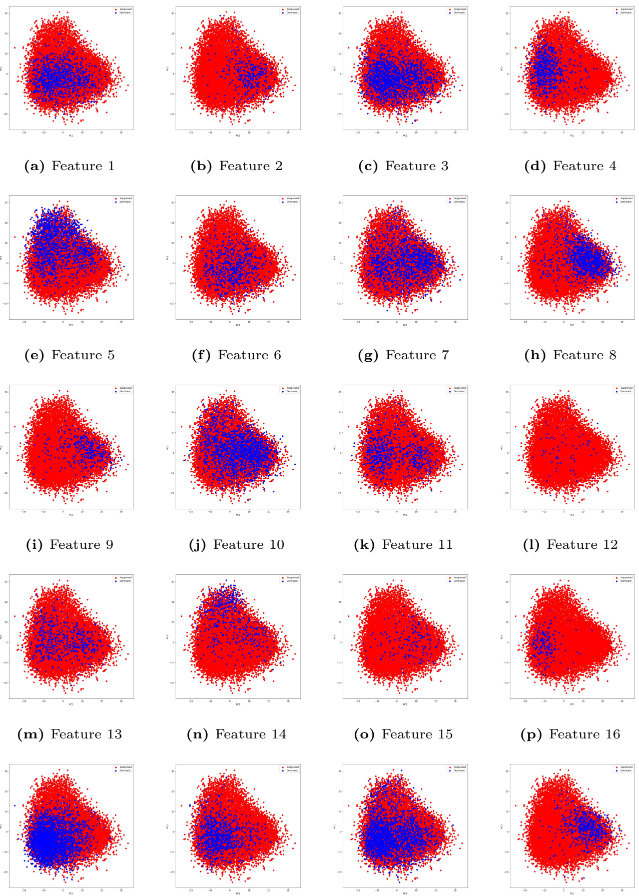
Author identification is one of the most relevant tasks artistic curation. Not only to properly identify the original painter of one artistic piece, but also to

Table 5: Performance measured using MCC for a GBC in the original and a balanced partition obtain by subsampling the original dataset.

	MCC performance	
Feature	Original partition	Balanced partition
1	0.08	0.26
2	0.00	0.52
3	0.10	0.30
4	0.03	0.43
5	0.23	0.48
6	0.02	0.25
7	0.05	0.25
8	0.06	0.54
9	0.10	0.40
10	0.07	0.21
11	0.07	0.37
12	0.00	0.34
13	0.00	0.18
14	0.20	0.56
15	0.00	0.19
16	0.00	0.60
17	0.22	0.42
18	0.05	0.34
19	0.06	0.20
20	0.00	0.56

Table 6: MCC obtained for each features using a FRBC.

Feature	MCC	Feature	MCC
1	0.2369	11	0.0000
2	0.5001	12	0.0000
3	0.2076	13	0.0954
4	0.2076	14	0.4666
5	0.4440	15	0.6888
6	0.1611	16	0.4714
7	0.1941	17	0.3708
8	0.4512	18	0.1478
9	0.3612	19	-0.2377
10	0.2432	20	0.4552



(a) Feature 1

(b) Feature 2

(c) Feature 3

(d) Feature 4

(e) Feature 5

(f) Feature 6

(g) Feature 7

(h) Feature 8

(i) Feature 9

(j) Feature 10

(k) Feature 11

(l) Feature 12

(m) Feature 13

(n) Feature 14

(o) Feature 15

(p) Feature 16

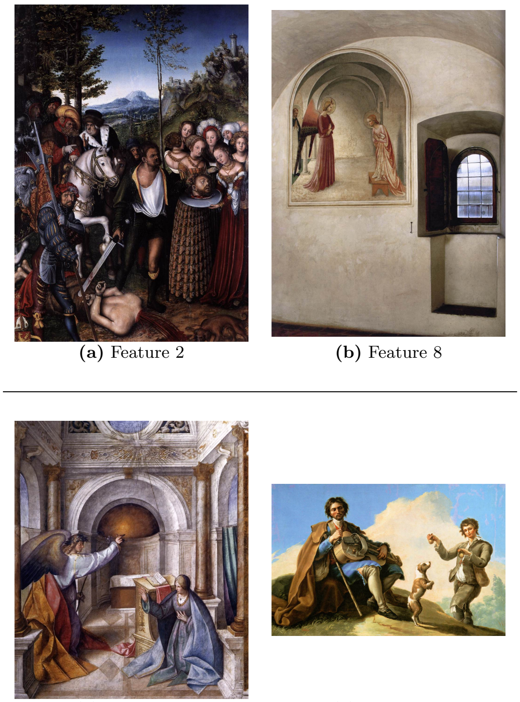
(q) Feature 17

(r) Feature 18

(s) Feature 19

(t) Feature 20

Figure 5: PCA projections for the deep features. Blue dots mark samples where the feature is dominant.



(a) Feature 2 **(b)** Feature 8

(a) Feature 15 **(a)** Feature 16

Figure 6: Example paintings for four of the dominant features studied. (a) “The behead of Saint John Baptist”, Caravaggio. (b) “The Annunciation and view from a Cell”, Fra Angelico. (c) “Annunciation to Mary”, Bocaccio (d) “The blind musician”, Ramón Bayeu y Subías.

Feature 2	DS
IF Early Renaissance IS High AND Northern Renaissance IS High	0.4968
Feature 8	
IF Early Renaissance IS High	0.4332
Feature 15	
IF Cubism IS Low AND Early Renaissance IS High AND Pointillism IS Low	0.1065
IF Early Renaissance IS High AND Rococo IS Low	0.3906
Feature 16	
IF Analytical Cubism IS Low AND Naive Art Primitivism IS Medium AND Pointillism IS Medium	0.2199
IF Contemporary Realism IS High AND Cubism IS Low AND High Renaissance IS Low	0.2932
IF Analytical Cubism IS Low AND Color Field Painting IS Medium AND Pop Art IS Low	0.1334

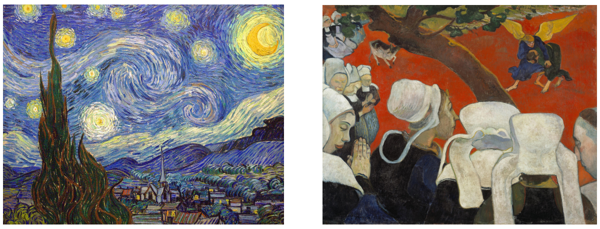
Table 7: Most important rules (DS >0.1) that identify some of the deep features studied.

detect possible forgeries or false attributions. Our previous deep learning models can output the likelihood of a paintings authorship, but these predictions tend to be overconfident and not realistic. In order to solve this problem, we use a a FRBC to distinguish between particular authors of interest, and output a more trustworthy estimation of the reliability of such prediction [22].

As a way of illustration of this application, we shall construct a FRBC to distinguish two painters that were acquainted in real life: Paul Gauguin and Vincent Van Gogh. Its is specially interesting to see if the resulting rules match the actual knowledge that we have of both of them. They are considered postimpresionist painters, with strong similarities and differences in their style (Figure 7).

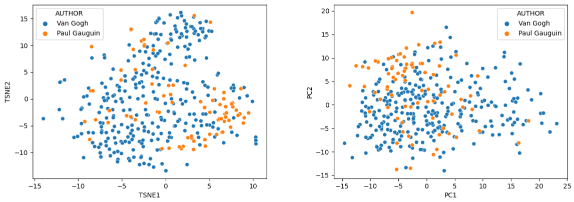
The SemArt contains 291 paintings from Van Gogh and 81 from Gauguin. Figure 8 shows a PCA and a TSN visualisation of the paintings for each class using our style and Grad-CAM characterization. We construct the rules using in the same way as in the previous section, but in this case the consequent class will be one of the two authors. Table 8 shows the performance for both the GBC and the FRBC. As the number of antecedents is too high for the genetic algorithm (see Section 5.3.1) to obtain a good result, we used the features that were more important using the GBC classifier to train the FRBC, which is also shown in Table 8. We obtained good results for both approaches, which again proves that the style and Grad-CAM heatmap characterisation was successful. Indeed, we even obtained better results with the FRBC compared to the GBC in the reduced set of features.

Figure 9 shows the rules obtained to differentiate both authors and the dominance score and individual accuracy for each rule in all the training samples they fired. We obtained three successful rules for Van Gogh and one for Gauiguin. From those rules we can see that Synthetic Cubism is a very good feature to identify Van Gogh paintings compared to Gauguin, and Early Renaissance style



(a) “Starry Night” **(b)** “Visions after the Sermon”

Figure 7: Vincent Van Gogh and Paul Gauguin. Van Gogh and Gauguin were both considered exceptional postimpresionist painters. Stetically, they both used a non-realistic style, brilliant colors and both were inspired by the compositions of japaneses stamps. However, they also present important differences in the content of their paintings. For example, Gauguin was interested profoundly in symbolism and was one of the pioneers that gave birth to this movement [30].



(a) **(b)**

Figure 8: Visualisation of Van Gogh and Gauguin samples using PCA (a) and TSNE (b). The features reduces are the style and Grad-CAM features.

the second best. We only found one relevant pattern for Gauguin. One interesting issue is that the best features to discriminate both artist are styles that did not exist in the actual time of the painters. This can indicate that these painters already started some of the traits that characterised those artistic movements.

6. Conclusions and future lines

In this paper we have proposed a new method to combine visual features and contextual annotations, using a BoW based on a fuzzy membership encoding, a FRBC approach and CLIP features. We used these methods in a classification framework, that considers a fine-tuned ResNet 50 enriched to extract the visual features from a dataset of artistic images. This network learns to solve a classification problem and to reconstruct the features extracted from the contextual

Table 8: Performance for a GBC and a FRBC in the Van Gogh/Gauguin identification task.

GBC		GBC-reduced features		FRBC	
Accuracy	MCC	Accuracy	MCC	Accuracy	MCC
1.00	1.00	0.88	0.53	0.88	0.62

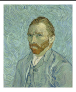
Author		Antecedents	DS	Train Acc	Test Acc
	1	IF New Realism IS Low AND Post Impressionism IS Medium	0.0076	0.5000	0.0000
	2	**IF Early Renaissance IS Medium AND New Realism IS Medium AND Synthetic Cubism IS Medium**	0.0740	0.7777	1.0000
	3	**IF Early Renaissance IS Low AND Synthetic Cubism IS High**	0.2517	0.9390	0.8888
	4	**IF Synthetic Cubism IS Low**	0.4624	0.9097	0.925
	5	IF Contemporary Realism IS Medium AND Synthetic Cubism IS Low AND Relevant area IS Low	0.0092	0.0000	0.0000
	6	**IF Contemporary Realism IS Medium AND Minimalism IS Low**	0.3389	0.7586	0.7692
	7	IF Early Renaissance IS Medium AND Minimalism IS Medium AND Synthetic Cubism IS Medium	0.0124	0.0000	0.0000

Figure 9: Rules that differentiate Van Gogh from Gauguin paintings.

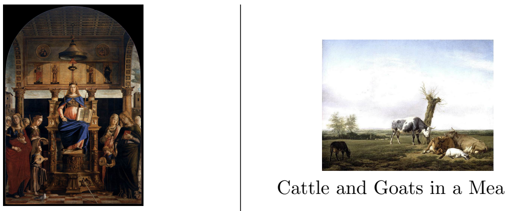
information for each image, which helps the network generalise better, as it does not need to rely only on visual cues to classify each sample. We have also studied how CLIP auto-encoder features compare in performance with models. Besides, we have proposed different alternatives to interpret the features obtained with these methods.

The comparison between contextual-aware models with some similar visual-only classification frameworks show favourable results for the latter ones. We obtained the best results overall using a MTL paradigm with contextual information. We also showed how some of the deep features used by the best model can be characterized according to the relevant parts of the image and the style of painting. Finally, we showed that some painters can be successfully characterised likewise and distinguish one from another also using interpretable patterns.

Future lines of our research shall study more expressive features in which to represent some of the images characteristics [41] and apply methods to improve the performance of the FRBC in imbalanced datasets. We also intend to develop a metric that can compute how good is a commentary that describes an image, so that the additional information present in it the text can be quantified.

7. Acknowledgements

This work was supported in part by Oracle Cloud credits and related resources provided by Oracle for Research and by MCIN/AEI/10.13039/501100011033 and ERDF “A way of making Europe” under grant CONFIA (PID2021-122916NB-I00).



St Veneranda Enthroned

The painting was originally in the church of Corpus Domini in Venice, where the body of the saint is preserved.

Cattle and Goats in a Meadow

The painting represents cattle and goats by a pollarded tree in a meadow, shepherd boys approaching beyond.

Figure A.10: Example of two contextual annotations in the dataset. The first one (a) talks about the context but does not mention anything about the contents of the painting itself. The other one (b) describes its contents.

Appendix A. SemArt Dataset and Context annotation study

For our experimentation we have used the SemArt dataset [29]. This dataset consists of 21,384 painting images. Following the original data partition in [29], 19,244 images are used for training (i.e. a 90%), 1,069 for validation, and 1,069 for test (i.e. a 5% each). Each painting has associated a textual artistic comment, alongside the following attributes: Author, Title, Date, Technique, Type, School, and Timeframe.

In this experimentation four different classification tasks are proposed:

- Type: each painting is classified according to 10 different common types of paintings: portrait, landscape, religious, etc.
- School: each painting is identified with different schools of art: Italian, Dutch, French, Spanish, etc. There are a total of 25 classes of this kind.
- Timeframe: The attribute Timeframe, which corresponds to periods of 50 years evenly distributed between 801 and 1900, is used to classify each painting according to its creation date. We consider only the timeframes where at least 10 paintings are present. This corresponds to 18 classes.
- Author: corresponds to the author of each paintings. We consider a total of 350 painters, that comprise the set of authors with more than 10 paintings in the dataset.

Appendix A.1. Encoding text annotations using Bag of Words and TF-IDF

The text annotation present in each of the paintings can be encoded in different ways. The most classical one is the Bag of Words model in which

Most Popular Words	
Painting	Right
Painted	Century
St	Life
Work	Scene
Picture	Christ
Figures	Painter
Left	Scenes
Portrait	Shows
Paintings	Panel
Artist	Church

Table A.9: Top 20 most common words found in the paintings contextual annotations.

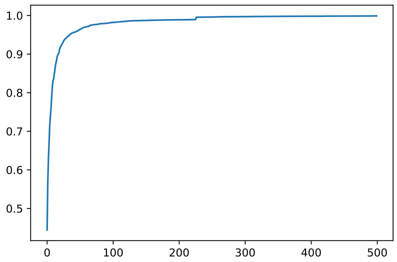
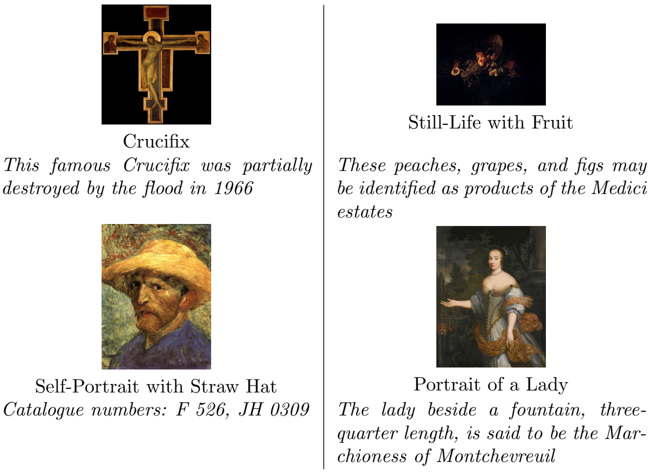


Figure A.11: Proportion of paintings where at least one of the values in the encoding vector is not 0 using the top-k most popular words in the dataset contextual annotations. The x axis represents the value of k.



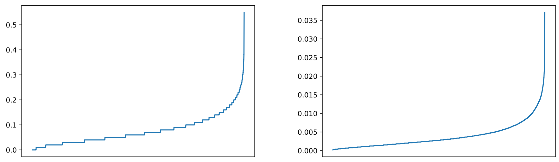
Crucifix
This famous Crucifix was partially destroyed by the flood in 1966

Still-Life with Fruit
These peaches, grapes, and figs may be identified as products of the Medici estates

Self-Portrait with Straw Hat
Catalogue numbers: F 526, JH 0309

Portrait of a Lady
The lady beside a fountain, three-quarter length, is said to be the Marchioness of Montchevreuil

Figure A.12: Examples of 4 paintings with contextual annotations that did not contain any of the 100 most common words in the dataset.



(a) 100 size encoding vector.

(b) 10445 size encoding vector.

Figure A.13: Proportion of values in the encoding vectors that are not 0 for each painting. Paintings sorted in the x axis according to a increasing permutation in this sense.

each phrase is represented as a vector of numbers in the $[0, 1]$ range where each position is associated with one word. Another popular representation is based on the TF-IDF metric representing each word [1], which weights the power of discrimination of each word in the whole document collection (artistic image database in our case). TF-IDF indexing has been very popular in other artistic image processing tasks [29] and in natural language processing and text mining tasks [48, 63].

TF-IDF metric indexing is more popular than BoW in modern applications. However, this representation presents some issues for our task. TF-IDF ponders most words that are frequent in a description but absent in others. The idea is that those words are most discriminative than others, but this is not necessarily true in our case. “Landscape”, for example, is a word that appears in many descriptions but is also very discriminative. This also happens with other words such as “Portrait”.

We have opted to compute both representations and compare their behaviour. In order to do so, we use as a vocabulary the list of words that appear more than 10 times in the contextual annotation and titles. We compute the encoding vectors using these words. We have also considered a simpler representation vector using BoW with only the 100 most popular words.

Most descriptions in the SemArt dataset contain contextual information about the painter and the painting, although some comments can be more descriptive than others about the actual visual information present in the canvas (Figure A.10). Table A.9 shows the most common words found in the paintings descriptions. Some of them are very general, like “painting” or “picture”. However, there are some religious words in the top too, like “Church” and “Christ”, suggesting that this theme is actually a popular topic in the whole collection of paintings.

Figure A.11 shows the percentage of the paintings that contain at least one of the most popular 500 words. Using the 100 most popular ones leaves all but 342 paintings with at least one word contained in that top. If we used the codification with the words that appeared more than 10 times, this results in the 100% with at least one symbol, as we are taking a total of $10,445$ words. If we wanted to use the top-k most popular words and we wanted all paintings to have at least one word contained in their description, we would need to use 3060 words.

Figure A.12 shows three contextual annotations that did not contained any of the considered words. We can see that these annotations are short, and contain specific words about the painting itself, like “peaches”, “Crucifix”, and “lady”, which are not as frequent as to appear in the 100 most common words. In other cases like in *Self-Portrait with Straw Hat* the contextual annotation is just an identifier in a catalogue.

The size of the $10,445$-word encoding vector might present a problem, because most of the entries will be 0, as the painting descriptions in this dataset are not long enough to present an important percentage of the $10,445$ possible words. Figure A.13 shows the percentage of words that appeared with respect to the vector size for both the 100-word vector encoding and the 10445-word

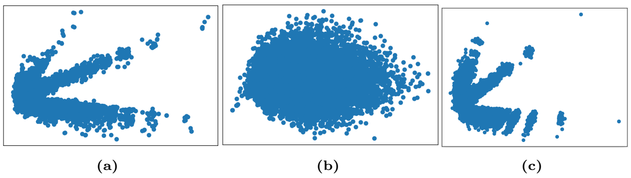


Figure A.14: PCA representation of the features computed using FCM on the BoW (a) and TF-IDF (b) using all the words that appear 10 times or more, and using BoW on the top 100 most common words (c).

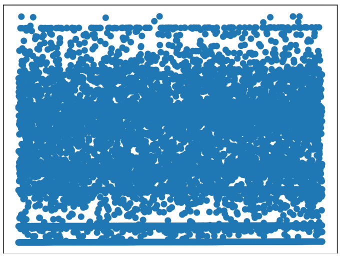


Figure A.15: PCA representation of the 40 fuzzy memberships computed using TF-IDF and FRBC fuzzy memberships, using the top 30 most common words.

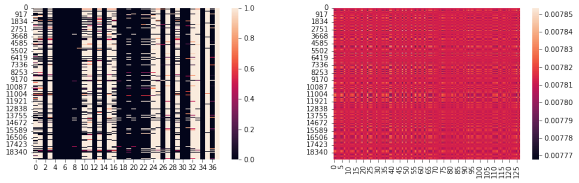
one. For the latter one, there is no painting with more than a 3.5% of the vector with values of 1. For the case of the 100-word encoding, we do not have the same problem.

Appendix A.2. Clustering structure on the extracted features

We tested both BoW and TF-IDF representations of the contextual annotations. For the number of clusters, we opted for 128 as it gave good results in the contextual embeddings of the ContextNet [28] and other works related to this net [13].

Figure A.14 illustrates the resulting FCM memberships using PCA with two components for the $10,445$-word and 100-word encodings. In this figure we can observe that the BoW features seem to have a more clear structure than TF-IDF representations, which should perform better in the classification task. In Figure A.15 we show the same result for the FRBC fuzzy memberships, which showed a very different structure from the FCM memberships.

Figure A.16 shows the fuzzy memberships using some of the different alternatives explored: using different BoW/TF-IDF models and using FCM or FRBC. We can see that the FCM has the problem that all the memberships must add 1, while the FRBC can surpass these limitation. We can also see that



(a) **(b)**

Figure A.16: Fuzzy memberships visualization for and FRBC with TF-IDF on the top 100 words.

in both cases most of the samples have high memberships to the same groups, which means that there is a big overlap between them.

Appendix B. Fuzzy rules based clustering

The modification of the original clustering algorithm includes two main changes:

1. The original algorithm gave as a result a crisp clustering. In order to return fuzzy memberships to the distinct groups, we use the value of the consequent for each rule selected in the algorithm.
2. The original algorithm had a stopping condition based on a stopping parameter, so that when a percentage of the original data was assigned to a group, it stopped. Since there is not a proper criteria to choose this parameter, we stop when all the original samples have been removed from the dataset.

The resulting algorithm is the following:

1. We designate the existing observations as the main data, $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$.
2. $z = 1$.
3. Generate a set of synthetic samples X'.
4. Compute the distance of the main data to their centroid, q, and the same for the synthetic data, q'.
5. If $q' < q$ return to step 3.
6. Generate a set of rules to discriminate between X and X'.
7. Select the rule with more data compatibility (Eq. **??** and Eq. **??**).

8. The membership of each observation to cluster z is the consequent of the chosen rule.

9. Remove the observations from X and X' that belonged to cluster z with more than 0.5 degree.

10. $z = z + 1$

11. If $\|X\| > 0$, return to step 3.

This approach has a advantage over the FCM. In the FCM the sum of all memberships must sum 1. This means that the bigger the contextual vector is, the lesser value each membership will be. In the case of the FRBC, there is not such restriction. Besides, we do not need to specify the number of clusters, as the process has a natural way to finish when all the observations have been assigned to a group.

In order to apply this algorithm, we need to use an algorithm to generate the set of rules for the classification problem (step 7 in Section 2.2). We have used the the LFL R package [10] to search for rules with 0.5 of minimum confidence, 0.02 of minimum support and a maximum number of 4 antecedents.

References

[1] Aizawa, A., 2003. An information-theoretic perspective of tf–idf measures. Information Processing & Management 39, 45–65.

[2] Aloysius, N., Geetha, M., 2017. A review on deep convolutional neural networks, in: 2017 International Conference on Communication and Signal Processing (ICCSP), IEEE. pp. 588–592.

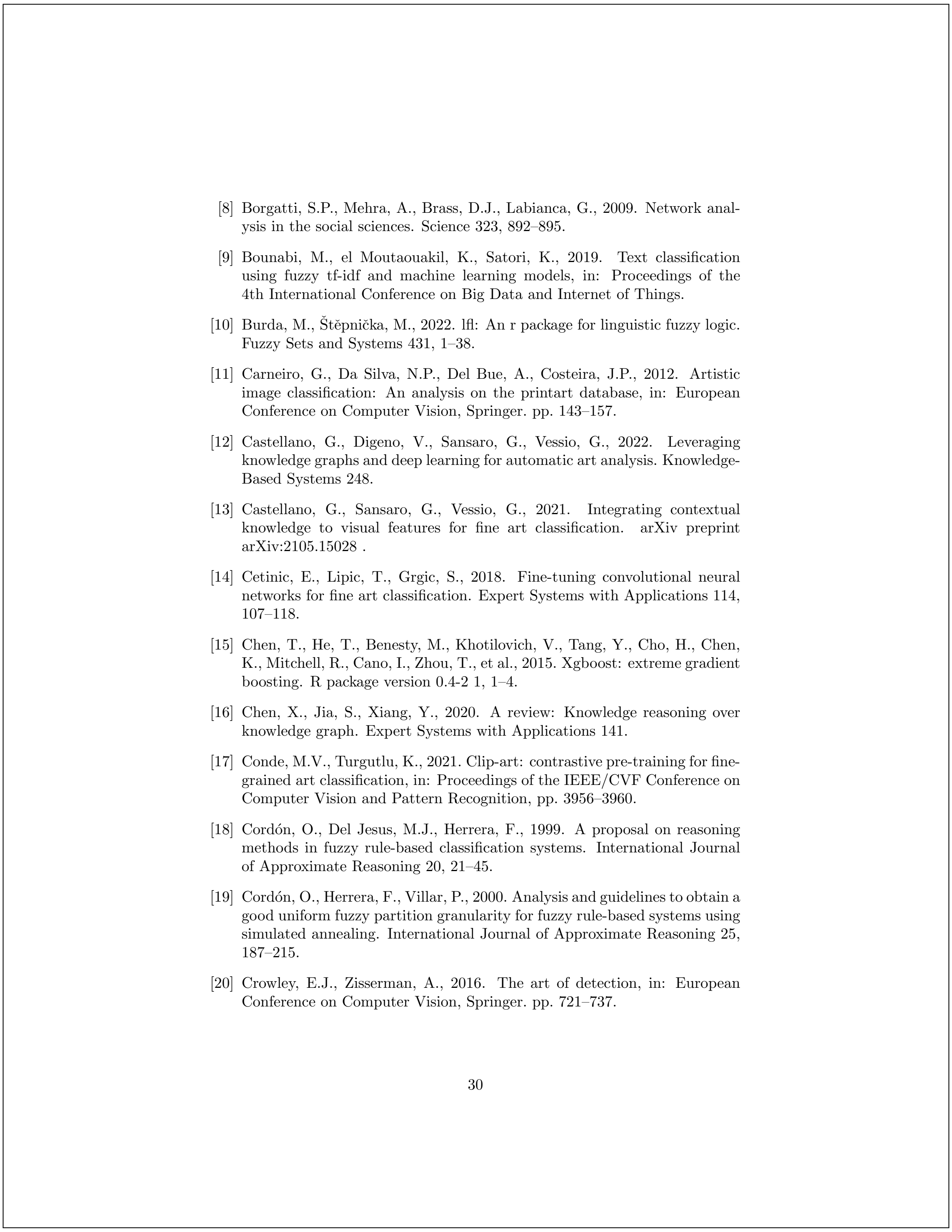
[3] Anwar, S.M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., Khan, M.K., 2018. Medical image analysis using convolutional neural networks: a review. Journal of Medical Systems 42, 1–13.

[4] Barni, M., Pelagotti, A., Piva, A., 2005. Image processing for the analysis and conservation of paintings: opportunities and challenges. IEEE Signal Processing Magazine 22, 141–144.

[5] Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1798–1828.

[6] Bezdek, J.C., Ehrlich, R., Full, W., 1984. FCM: The fuzzy c-means clustering algorithm. Computers & Geosciences 10, 191–203.

[7] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E., 2008. Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008.



[8] Borgatti, S.P., Mehra, A., Brass, D.J., Labianca, G., 2009. Network analysis in the social sciences. Science 323, 892–895.

[9] Bounabi, M., el Moutaouakil, K., Satori, K., 2019. Text classification using fuzzy tf-idf and machine learning models, in: Proceedings of the 4th International Conference on Big Data and Internet of Things.

[10] Burda, M., Štěpnička, M., 2022. lfl: An r package for linguistic fuzzy logic. Fuzzy Sets and Systems 431, 1–38.

[11] Carneiro, G., Da Silva, N.P., Del Bue, A., Costeira, J.P., 2012. Artistic image classification: An analysis on the printart database, in: European Conference on Computer Vision, Springer. pp. 143–157.

[12] Castellano, G., Digeno, V., Sansaro, G., Vessio, G., 2022. Leveraging knowledge graphs and deep learning for automatic art analysis. Knowledge-Based Systems 248.

[13] Castellano, G., Sansaro, G., Vessio, G., 2021. Integrating contextual knowledge to visual features for fine art classification. arXiv preprint arXiv:2105.15028 .

[14] Cetinic, E., Lipic, T., Grgic, S., 2018. Fine-tuning convolutional neural networks for fine art classification. Expert Systems with Applications 114, 107–118.

[15] Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., et al., 2015. Xgboost: extreme gradient boosting. R package version 0.4-2 1, 1–4.

[16] Chen, X., Jia, S., Xiang, Y., 2020. A review: Knowledge reasoning over knowledge graph. Expert Systems with Applications 141.

[17] Conde, M.V., Turgutlu, K., 2021. Clip-art: contrastive pre-training for fine-grained art classification, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3956–3960.

[18] Cordón, O., Del Jesus, M.J., Herrera, F., 1999. A proposal on reasoning methods in fuzzy rule-based classification systems. International Journal of Approximate Reasoning 20, 21–45.

[19] Cordón, O., Herrera, F., Villar, P., 2000. Analysis and guidelines to obtain a good uniform fuzzy partition granularity for fuzzy rule-based systems using simulated annealing. International Journal of Approximate Reasoning 25, 187–215.

[20] Crowley, E.J., Zisserman, A., 2016. The art of detection, in: European Conference on Computer Vision, Springer. pp. 721–737.

[21] Elgammal, A., Liu, B., Kim, D., Elhoseiny, M., Mazzone, M., 2018. The shape of art history in the eyes of the machine, in: Proceedings of the AAAI Conference on Artificial Intelligence.

[22] Fernández, G., Aledo, J.A., Gamez, J.A., Puerta, J.M., 2022. Factual and counterfactual explanations in fuzzy classification trees. IEEE Transactions on Fuzzy Systems 30, 5484–5495. doi:10.1109/TFUZZ.2022.3179582.

[23] Fumanal-Idocin, J., Alonso-Betanzos, A., Cordón, O., Bustince, H., Minárová, M., 2020. Community detection and social network analysis based on the italian wars of the 15th century. Future Generation Computer Systems 113, 25 – 40.

[24] Fumanal-Idocin, J., Cordón, O., Bustince, H., 2023a. The krypteia ensemble: Designing classifier ensembles using an ancient spartan military tradition. Information Fusion 90, 283–297.

[25] Fumanal-Idocin, J., Cordón, O., Dimuro, G.P., López-de Hierro, A.F.R., Bustince, H., 2023b. Quantifying external information in social network analysis: An application to comparative mythology. IEEE Transactions on Cybernetics , 1–0doi:10.1109/TCYB.2023.3239555.

[26] Fumanal-Idocin, J., Takáč, Z., Horanská, L., Bustince, H., Cordon, O., 2022. Fuzzy clustering to encode contextual information in artistic image classification, in: Information Processing and Management of Uncertainty in Knowledge-Based Systems, Springer International Publishing. pp. 355–366.

[27] Garcia, N., Renoust, B., Nakashima, Y., 2019. Context-aware embeddings for automatic art analysis, in: Proceedings of the 2019 on International Conference on Multimedia Retrieval, pp. 25–33.

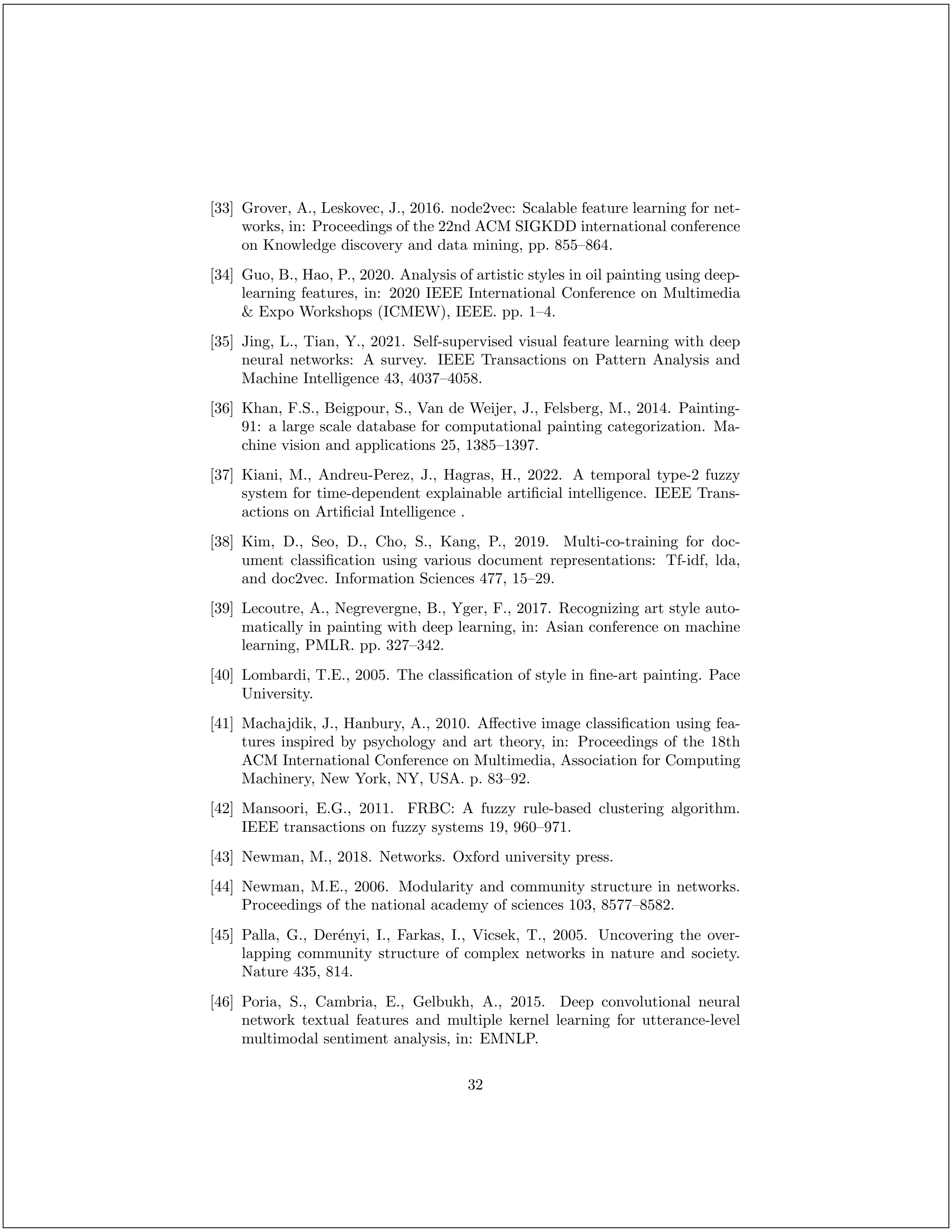
[28] Garcia, N., Renoust, B., Nakashima, Y., 2020. Contextnet: representation and exploration for painting classification and retrieval in context. International Journal of Multimedia Information Retrieval 9, 17–30.

[29] Garcia, N., Vogiatzis, G., 2018. How to read paintings: semantic art understanding with multi-modal retrieval, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops.

[30] Gombrich, E.H., Gombrich, E.H., 1995. The story of art. volume 12. Phaidon London.

[31] Gonthier, N., Gousseau, Y., Ladjal, S., Bonfait, O., 2018. Weakly supervised object detection in artworks, in: Proceedings of the European Conference on Computer Vision (ECCV) Workshops, pp. 1–18.

[32] Grohe, M., 2020. word2vec, node2vec, graph2vec, x2vec: Towards a theory of vector embeddings of structured data, in: Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, pp. 1–16.



[33] Grover, A., Leskovec, J., 2016. node2vec: Scalable feature learning for networks, in: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855–864.

[34] Guo, B., Hao, P., 2020. Analysis of artistic styles in oil painting using deep-learning features, in: 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), IEEE. pp. 1–4.

[35] Jing, L., Tian, Y., 2021. Self-supervised visual feature learning with deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 43, 4037–4058.

[36] Khan, F.S., Beigpour, S., Van de Weijer, J., Felsberg, M., 2014. Painting-91: a large scale database for computational painting categorization. Machine vision and applications 25, 1385–1397.

[37] Kiani, M., Andreu-Perez, J., Hagras, H., 2022. A temporal type-2 fuzzy system for time-dependent explainable artificial intelligence. IEEE Transactions on Artificial Intelligence .

[38] Kim, D., Seo, D., Cho, S., Kang, P., 2019. Multi-co-training for document classification using various document representations: Tf-idf, lda, and doc2vec. Information Sciences 477, 15–29.

[39] Lecoutre, A., Negrevergne, B., Yger, F., 2017. Recognizing art style automatically in painting with deep learning, in: Asian conference on machine learning, PMLR. pp. 327–342.

[40] Lombardi, T.E., 2005. The classification of style in fine-art painting. Pace University.

[41] Machajdik, J., Hanbury, A., 2010. Affective image classification using features inspired by psychology and art theory, in: Proceedings of the 18th ACM International Conference on Multimedia, Association for Computing Machinery, New York, NY, USA. p. 83–92.

[42] Mansoori, E.G., 2011. FRBC: A fuzzy rule-based clustering algorithm. IEEE transactions on fuzzy systems 19, 960–971.

[43] Newman, M., 2018. Networks. Oxford university press.

[44] Newman, M.E., 2006. Modularity and community structure in networks. Proceedings of the national academy of sciences 103, 8577–8582.

[45] Palla, G., Derényi, I., Farkas, I., Vicsek, T., 2005. Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814.

[46] Poria, S., Cambria, E., Gelbukh, A., 2015. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis, in: EMNLP.

[47] Pu, Y., Gan, Z., Henao, R., Yuan, X., Li, C., Stevens, A., Carin, L., 2016. Variational autoencoder for deep learning of images, labels and captions, in: NIPS.

[48] Qaiser, S., Ali, R., 2018. Text mining: use of tf-idf to examine the relevance of words to documents. International Journal of Computer Applications 181, 25–29.

[49] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR. pp. 8748–8763.

[50] Rawat, W., Wang, Z., 2017. Deep convolutional neural networks for image classification: A comprehensive review. Neural computation 29, 2352–2449.

[51] Saleh, B., Elgammal, A., 2015. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. arXiv preprint arXiv:1505.00855 .

[52] Sanz, J.A., Fernández, A., Bustince, H., Herrera, F., 2013. Ivturs: A linguistic fuzzy rule-based classification system based on a new interval-valued fuzzy reasoning method with tuning and rule selection. IEEE Transactions on Fuzzy Systems 21, 399–411.

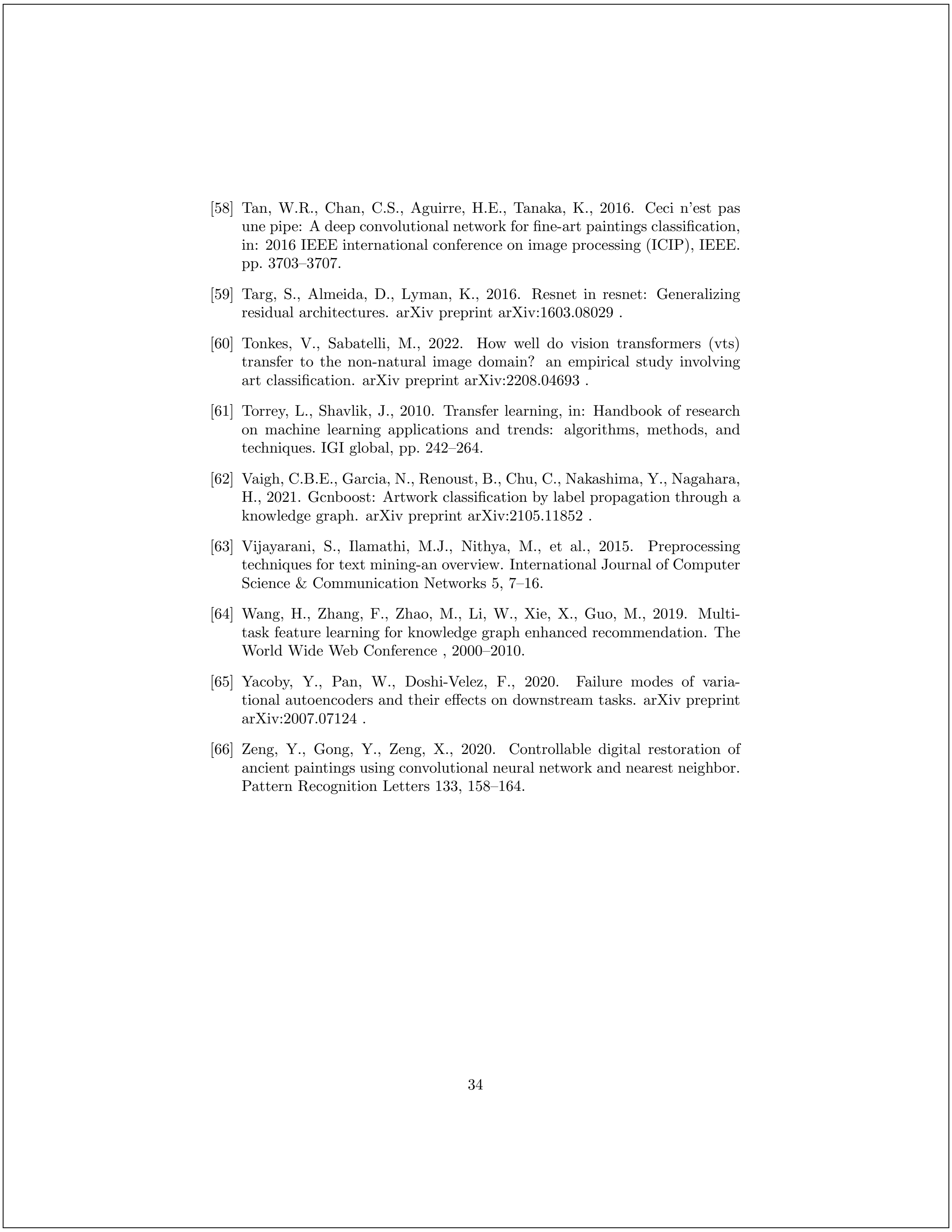
[53] Sarlin, P.E., DeTone, D., Malisiewicz, T., Rabinovich, A., 2020. Superglue: Learning feature matching with graph neural networks. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) , 4937–4946.

[54] Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, pp. 618–626.

[55] Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition, in: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

[56] Škrjanc, I., Iglesias, J.A., Sanchis, A., Leite, D., Lughofer, E., Gomide, F., 2019. Evolving fuzzy and neuro-fuzzy approaches in clustering, regression, identification, and classification: A survey. Information sciences 490, 344–368.

[57] Taber, R., 1991. Knowledge processing with fuzzy cognitive maps. Expert systems with applications 2, 83–87.



[58] Tan, W.R., Chan, C.S., Aguirre, H.E., Tanaka, K., 2016. Ceci n'est pas une pipe: A deep convolutional network for fine-art paintings classification, in: 2016 IEEE international conference on image processing (ICIP), IEEE. pp. 3703–3707.

[59] Targ, S., Almeida, D., Lyman, K., 2016. Resnet in resnet: Generalizing residual architectures. arXiv preprint arXiv:1603.08029 .

[60] Tonkes, V., Sabatelli, M., 2022. How well do vision transformers (vts) transfer to the non-natural image domain? an empirical study involving art classification. arXiv preprint arXiv:2208.04693 .

[61] Torrey, L., Shavlik, J., 2010. Transfer learning, in: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI global, pp. 242–264.

[62] Vaigh, C.B.E., Garcia, N., Renoust, B., Chu, C., Nakashima, Y., Nagahara, H., 2021. Gcnboost: Artwork classification by label propagation through a knowledge graph. arXiv preprint arXiv:2105.11852 .

[63] Vijayarani, S., Ilamathi, M.J., Nithya, M., et al., 2015. Preprocessing techniques for text mining-an overview. International Journal of Computer Science & Communication Networks 5, 7–16.

[64] Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., Guo, M., 2019. Multi-task feature learning for knowledge graph enhanced recommendation. The World Wide Web Conference , 2000–2010.

[65] Yacoby, Y., Pan, W., Doshi-Velez, F., 2020. Failure modes of variational autoencoders and their effects on downstream tasks. arXiv preprint arXiv:2007.07124 .

[66] Zeng, Y., Gong, Y., Zeng, X., 2020. Controllable digital restoration of ancient paintings using convolutional neural network and nearest neighbor. Pattern Recognition Letters 133, 158–164.

References

[1] Dhanesh Ramachandram and Graham W. Taylor. **Deep Multimodal Learning: A Survey on Recent Advances and Trends**. *IEEE Signal Processing Magazine*, **34**:96–108, 2017.

[2] Tong Meng, Xuyang Jing, Zheng Yan, and Witold Pedrycz. **A survey on machine learning for data fusion**. *Information Fusion*, **57**:115–129, 2020.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. **Learning transferable visual models from natural language supervision**. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[4] Gleb Beliakov, Humberto Bustince Sola, and Tomasa Calvo Sánchez. *A practical guide to averaging functions*, **329**. Springer, 2016.

[5] Michel Grabisch, Jean-Luc Marichal, Radko Mesiar, and Endre Pap. *Aggregation functions*, **127**. Cambridge University Press, 2009.

[6] Giancarlo Lucca, José Antonio Sanz, Graçaliz Pereira Dimuro, Benjamin Bedregal, Radko Mesiar, Anna Kolesarova, and Humberto Bustince. **Preaggregation functions: construction and an application**. *IEEE Transactions on Fuzzy Systems*, **24**(2):260–272, 2015.

[7] Jabeen Summaira, Xi Li, Amin Muhammad Shoib, Songyuan Li, and Jabbar Abdul. **Recent Advances and Trends in Multimodal Deep Learning: A Review**. *ArXiv*, **abs/2105.11087**, 2021.

[8] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. **Deep learning**. *nature*, **521**(7553):436–444, 2015.

[9] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. **Learning Spatiotemporal Features with 3D Convolutional Networks**. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497, 2015.

[10] Fabien Lotte and Cuntai Guan. **Regularizing Common Spatial Patterns to Improve BCI Designs: Unified Theory and New Algorithms**. *IEEE Transactions on Biomedical Engineering*, **58**:355–362, 2011.

[11] C Vidaurre, C Klauer, T Schauer, A Ramos-Murguialday, and K-R Müller. **EEG-based BCI for the linear control of an upper-limb neuroprosthesis**. *Medical engineering & physics*, **38**:1195–1204, 2016.

[12] C. Vidaurre, A. Ramos Murguialday, S. Haufe, M. Gómez, K.-R. Müller, and V.V. Nikulin. **Enhancing sensorimotor BCI performance with assistive afferent activity: An online evaluation**. *NeuroImage*, **199**:375 – 386, 2019.

[13] Mark Newman. *Networks*. Oxford university press, 2018.

[14] David Kempe, Jon M. Kleinberg, and Éva Tardos. **Maximizing the spread of influence through a social network**. In *KDD '03*, 2003.

[15] Wei Chen, Chi Wang, and Yajun Wang. **Scalable influence maximization for prevalent viral marketing in large-scale social networks**. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2010.

[16] Alex Fornito, Andrew Zalesky, and Edward Bullmore. *Fundamentals of brain network analysis*. Academic Press, 2016.

[17] Zhen Zhang, Hongxia Yang, Jiajun Bu, Sheng Zhou, Pinggang Yu, Jianwei Zhang, Martin Ester, and Can Wang. **ANRL: attributed network representation learning via deep neural networks.** In *Ijcai*, **18**, pages 3155–3161, 2018.

[18] Xiao Huang, Jundong Li, and Xia Hu. **Accelerated attributed network embedding**. In *Proceedings of the 2017 SIAM international conference on data mining*, pages 633–641. SIAM, 2017.

[19] Aditya Grover and Jure Leskovec. **node2vec: Scalable feature learning for networks**. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864, 2016.

[20] Chunyan Xu, Canyi Lu, Xiaodan Liang, Junbin Gao, Wei Zheng, Tianjiang Wang, and Shuicheng Yan. **Multi-loss Regularized Deep Neural Network**. *IEEE Transactions on Circuits and Systems for Video Technology*, **26**:2273–2283, 2016.

[21] Noa Garcia, Benjamin Renoust, and Yuta Nakashima. **Context-aware embeddings for automatic art analysis**. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval*, pages 25–33, 2019.

[22] Sinno Jialin Pan and Qiang Yang. **A Survey on Transfer Learning**. *IEEE Transactions on Knowledge and Data Engineering*, **22**:1345–1359, 2010.

[23] Jianping Gou, B. Yu, Stephen J. Maybank, and Dacheng Tao. **Knowledge Distillation: A Survey**. *Int. Journal of Computer Vision*, **129**:1789–1819, 2021.

[24] Abdulrahman H Altalhi, Juan I Forcén, Miguel Pagola, E Barrenechea, Humberto Bustince, and Zdenko Takáč. **Moderate deviation and restricted equivalence functions for measuring similarity between data**. *Information Sciences*, **501**:19–29, 2019.

[25] Humberto Bustince, Gleb Beliakov, Gracaliz Pereira Dimuro, Benjamin Bedregal, and Radko Mesiar. **On the definition of penalty functions in data aggregation**. *Fuzzy Sets and Systems*, **323**:1–18, 2017.

[26] Humberto Bustince, Edurne Barrenechea, Miguel Pagola, Javier Fernandez, Zeshui Xu, Benjamin Bedregal, Javier Montero, Hani Hagras, Francisco Herrera, and Bernard De Baets. **A historical account of types of fuzzy sets and their relationships**. *IEEE Transactions on Fuzzy Systems*, **24**(1):179–194, 2015.

[27] Miller McPherson, Lynn Smith-Lovin, and James M Cook. **Birds of a feather: Homophily in social networks**. *Annual review of sociology*, pages 415–444, 2001.

[28] Sergio Currarini, Jesse Matheson, and Fernando Vega-Redondo. **A simple model of homophily in social networks**. *European Economic Review*, **90**:18–39, 2016.

[29] Tao Wang, Hongjue Wang, and Xiaoxia Wang. **A novel cosine distance for detecting communities in complex networks**. *Physica A: Statistical Mechanics and its Applications*, **437**:21–35, 2015.

[30] Erjia Yan and Ying Ding. **Scholarly network similarities: How bibliographic coupling networks, citation networks, cocitation networks, topical networks, coauthorship networks, and coword networks relate to each other**. *Journal of the American Society for Information Science and Technology*, **63**(7):1313–1326, 2012.

[31] Gert Pfurtscheller, Christa Neuper, GR Muller, Bernhard Obermaier, Gunter Krausz, A Schlogl, Reinhold Scherer, Bernhard Graimann, Claudia Keinrath, Dimitris Skliris, et al. **Graz-BCI: state of the art and clinical applications**. *IEEE Transactions on neural systems and rehabilitation engineering*, **11**(2):1–4, 2003.

[32] Eoin Thomas, Matthew Dyson, and Maureen Clerc. **An analysis of performance evaluation for motor-imagery based BCI**. *Journal of neural engineering*, **10**(3):031001, 2013.

[33] Dana Lahat, Tülay Adali, and Christian Jutten. **Multimodal data fusion: an overview of methods, challenges, and prospects**. *Proceedings of the IEEE*, **103**(9):1449–1477, 2015.

[34] Sigrid Norris. *Systematically working with multimodal data: Research methods in multimodal discourse analysis*. John Wiley & Sons, 2019.

[35] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. **Deep learning scaling is predictable, empirically**. *arXiv preprint arXiv:1712.00409*, 2017.

[36] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. **Zero-shot text-to-image generation**. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[37] Mikel Galar, Alberto Fernández, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. **An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes**. *Pattern Recognition*, **44**(8):1761–1776, 2011.

[38] Mikel Galar, Alberto Fernandez, Edurne Barrenechea, Humberto Bustince, and Francisco Herrera. **A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches**. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **42**(4):463–484, 2011.

[39] Azlinah Mohamed, Maryam Khanian Najafabadi, Yap Bee Wah, Ezzatul Akmal Kamaru Zaman, and Ruhaila Maskat. **The state of the art and taxonomy of big data analytics: view from new big data framework**. *Artificial Intelligence Review*, **53**(2):989–1037, 2020.

[40] Antonio Galicia, R Talavera-Llames, A Troncoso, Irena Koprinska, and Francisco Martínez-Álvarez. **Multi-step forecasting for big data time series based on ensemble learning**. *Knowledge-Based Systems*, **163**:830–841, 2019.

[41] Frank Emmert-Streib, Zhen Yang, Han Feng, Shailesh Tripathi, and Matthias Dehmer. **An introductory review of deep learning for prediction models with big data**. *Frontiers in Artificial Intelligence*, **3**:4, 2020.

[42] Shazia Tabassum, Fabiola SF Pereira, Sofia Fernandes, and João Gama. **Social network analysis: An overview**. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **8**(5):e1256, 2018.

[43] Fang Wang, Menggang Li, Yiduo Mei, and Wenrui Li. **Time series data mining: A case study with big data analytics approach**. *IEEE Access*, **8**:14322–14328, 2020.

[44] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. **Grad-cam: Visual explanations from deep networks via gradient-based localization**. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[45] Si Zhang, Hanghang Tong, Jiejun Xu, and Ross Maciejewski. **Graph convolutional networks: a comprehensive review**. *Computational Social Networks*, **6**(1):1–23, 2019.

[46] Hani Hagras. **Toward Human-Understandable, Explainable AI**. *Computer*, **51**(9):28–36, 2018.

[47] Iosu Rodriguez-Martinez, Julio Lafuente, Regivan H.N. Santiago, Graçaliz Pereira Dimuro, Francisco Herrera, and Humberto Bustince. **Replacing pooling functions in Convolutional Neural Networks by linear combinations of increasing functions**. *Neural Networks*, **152**:380–393, 2022.

[48] H Bustince, E Barrenechea, Tomasa Calvo, Simon James, and Gleb Beliakov. **Consensus in multi-expert decision making problems using penalty functions defined over a Cartesian product of lattices**. *Information Fusion*, **17**:56–64, 2014.

[49] Mikel Elkano, Mikel Galar, Jose Sanz, and Humberto Bustince. **CHI-BD: A fuzzy rule-based classification system for Big Data classification problems**. *Fuzzy Sets and Systems*, **348**:75–101, 2018.

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

[51] Yacine Boudria, Amal Feltane, and Walter Besio. **Significant improvement in one-dimensional cursor control using Laplacian electroencephalography over electroencephalography**. *Journal of neural engineering*, **11**(3):035014, 2014.

[52] Claudio Carvalhaes and J Acacio De Barros. **The surface Laplacian technique in EEG: Theory and methods**. *International Journal of Psychophysiology*, **97**(3):174–188, 2015.

[53] Kai Keng Ang, Zheng Yang Chin, Haihong Zhang, and Cuntai Guan. **Filter bank common spatial pattern (FBCSP) in brain-computer interface**. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 2390–2397. IEEE, 2008.

[54] Laura De Miguel, Daniel Gómez, J Tinguaro Rodríguez, Javier Montero, Humberto Bustince, Graçaliz P Dimuro, and José Antonio Sanz. **General overlap functions**. *Fuzzy Sets and Systems*, **372**:81–96, 2019.

[55] Graçaliz Pereira Dimuro, Javier Fernández, Benjamin Bedregal, Radko Mesiar, José Antonio Sanz, Giancarlo Lucca, and Humberto Bustince. **The state-of-art of the generalizations of the Choquet integral: from aggregation and pre-aggregation to ordered directionally monotone functions**. *Information Fusion*, **57**:27–43, 2020.

[56] Francesco Bardozzo, Borja De La Osa, L'ubomíra Horanská, Javier Fumanal-Idocin, Mattia delli Priscoli, Luigi Troiano, Roberto Tagliaferri, Javier Fernandez, and Humberto Bustince. **Sugeno integral generalization applied to improve adaptive image binarization**. *Information Fusion*, **68**:37–45, 2021.

[57] Ronald R Yager. **Families of OWA operators**. *Fuzzy sets and systems*, **59**(2):125–148, 1993.

[58] Jeffrey P Nguyen, Frederick B Shipley, Ashley N Linder, George S Plummer, Mochi Liu, Sagar U Setru, Joshua W Shaevitz, and Andrew M Leifer. **Whole-brain calcium imaging with cellular resolution in freely behaving Caenorhabditis elegans**. *Proceedings of the National Academy of Sciences*, **113**(8):E1074–E1081, 2016.

[59] Haihong Zhang, Cuntai Guan, Kai Keng Ang, and Zheng Yang Chin. **BCI competition IV–data set I: learning discriminative patterns for self-paced EEG-based motor imagery detection**. *Frontiers in neuroscience*, **6**:7, 2012.

[60] Pradeep Kumar Mishra, B Jagadish, MPRS Kiran, Pachamuthu Rajalakshmi, and D Santhosh Reddy. **A novel classification for EEG based four class motor imagery using kullback-leibler regularized Riemannian manifold**. In *2018 IEEE 20th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–5. IEEE, 2018.

[61] S. Selim, M. M. Tantawi, H. A. Shedeed, and A. Badr. **A CSP AM-BA-SVM Approach for Motor Imagery BCI System**. *IEEE Access*, **6**:49192–49208, 2018.

[62] Sara Razi, Mohammad Reza Karami Mollaei, and Jamal Ghasemi. **A novel method for classification of BCI multi-class motor imagery task based on Dempster–Shafer theory**. *Information Sciences*, **484**:14 – 26, 2019.

[63] Aranzazu Jurio, Miguel Pagola, Radko Mesiar, Gleb Beliakov, and Humberto Bustince. **Image magnification using interval information**. *IEEE Transactions on Image Processing*, **20**(11):3112–3123, 2011.

[64] Clemens Brunner, Muhammad Naeem, Robert Leeb, Bernhard Graimann, and Gert Pfurtscheller. **Spatial filtering and selection of optimized components in four class motor imagery EEG data using independent components analysis**. *Pattern recognition letters*, **28**(8):957–964, 2007.

[65] H Bustince, E Barrenechea, and Miguel Pagola. **Restricted equivalence functions**. *Fuzzy Sets and Systems*, **157**(17):2333–2346, 2006.

[66] M. Hersche, T. Rellstab, P. D. Schiavone, L. Cavigelli, L. Benini, and A. Rahimi. **Fast and Accurate Multiclass Inference for MI-BCIs Using Large Multiscale Temporal and Spectral Features**. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 1690–1694, 2018.

[67] Anirban Chowdhury and Javier Andreu-Perez. **Clinical Brain-Computer Interface Challenge 2020 (CBCIC at WCCI2020): Overview, methods and results**. *IEEE Transactions on Medical Robotics and Bionics*, 2021.

[68] W.E. Wright. **Gravitational clustering**. *Pattern Recognition*, **9**(3):151 – 166, 1977.

[69] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera. **KEEL: a software tool to assess evolutionary algorithms for data mining problems**. *Soft Computing*, **13**(3):307–318, Feb 2009.

[70] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. **LOF: identifying density-based local outliers**. In *ACM sigmod record*, **29**, pages 93–104. ACM, 2000.

[71] Kun-Lun Li, Hou-Kuan Huang, Sheng-Feng Tian, and Wei Xu. **Improving one-class SVM for anomaly detection**. In *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*, **5**, pages 3077–3081. IEEE, 2003.

[72] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. **Isolation forest**. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

[73] José Catalàn Deus. **El príncipe del Renacimiento**. *Debate*, **26**:90, 2008.

[74] Mark EJ Newman and Michelle Girvan. **Finding and evaluating community structure in networks**. *Physical review E*, **69**(2):026113, 2004.

[75] Mark EJ Newman. **Fast algorithm for detecting community structure in networks**. *Physical review E*, **69**(6):066133, 2004.

[76] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. **Fast unfolding of communities in large networks**. *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10), oct 2008.

[77] Mark EJ Newman. **Finding community structure in networks using the eigenvectors of matrices**. *Physical review E*, **74**(3):036104, 2006.

[78] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. **Near linear time algorithm to detect community structures in large-scale networks**. *Physical review E*, **76**(3):036106, 2007.

[79] XingTing Pu, Radko Mesiar, Ronald R Yager, and LeSheng Jin. **Interval Sugeno integral with preference**. *IEEE Transactions on Fuzzy Systems*, **28**(3):597–601, 2019.

[80] J Fumanal-Idocin, A Alonso-Betanzos, O Cordón, H Bustince, and M Minárová. **Community detection and social network analysis based on the Italian wars of the 15th century**. *Future Generation Computer Systems*, **113**:25 – 40, 2020.

[81] Xiaohua Shi, Hongtao Lu, Yangchen He, and Shan He. **Community detection in social network with pairwisely constrained symmetric non-negative matrix factorization**. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 541–546, 2015.

[82] Yenumula B Reddy. **Role of game models in social networks**. In *2009 International Conference on Computational Science and Engineering*, **4**, pages 1131–1136. IEEE, 2009.

[83] James George Frazer. *The Golden Bough*. Palgrave Macmillan UK, 1922.

[84] Juan Eduardo Cirlot. *Diccionario de símbolos (in Spanish)*. Siruela, 2004.

[85] Scott S Elliott. *Reinventing Religious Studies: Key Writings in the History of a Discipline*. Routledge, 2014.

[86] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. **ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software**. *PloS One*, **9**(6):e98679, 2014.

[87] Andrew J Bayliss. *The Spartans*. Oxford University Press, 2020.

[88] Dylan Futter. **Plutarch, Plato and Sparta**. *Akroterion*, **57**(1):35–51, 2012.

[89] Javier Fumanal-Idocin, Yu-Kai Wang, Chin-Teng Lin, Javier Fernández, José Antonio Sanz, and Humberto Bustince. **Motor-Imagery-Based Brain-Computer Interface Using Signal Derivation and Aggregation Functions**. *IEEE Transactions on Cybernetics*, 2021.

[90] Yoav Freund and Robert E Schapire. **A decision-theoretic generalization of on-line learning and an application to boosting**. *Journal of Computer and System Sciences*, **55**(1):119–139, 1997.

[91] Leo Breiman. **Bagging predictors**. *Machine learning*, **24**(2):123–140, 1996.

[92] Leo Breiman. **Random forests**. *Machine learning*, **45**(1):5–32, 2001.

[93] Albert HR Ko, Robert Sabourin, and Alceu Souza Britto Jr. **From dynamic classifier selection to dynamic ensemble selection**. *Pattern recognition*, **41**(5):1718–1731, 2008.

[94] Salvador García, Zhong-Liang Zhang, Abdulrahman Altalhi, Saleh Alshomrani, and Francisco Herrera. **Dynamic ensemble selection for multi-class imbalanced datasets**. *Information Sciences*, **445**:22–37, 2018.

[95] Tomasz Woloszynski and Marek Kurzynski. **A probabilistic model of classifier competence for dynamic ensemble selection**. *Pattern Recognition*, **44**(10-11):2656–2668, 2011.

[96] Rafael MO Cruz, Robert Sabourin, George DC Cavalcanti, and Tsang Ing Ren. **META-DES: A dynamic ensemble selection framework using meta-learning**. *Pattern Recognition*, **48**(5):1925–1935, 2015.

[97] Tianqi Chen and Carlos Guestrin. **Xgboost: A scalable tree boosting system**. In *Proceedings of the 22nd ACM sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[98] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. **A comparative analysis of gradient boosting algorithms**. *Artificial Intelligence Review*, **54**(3):1937–1967, 2021.

[99] Noa Garcia, Benjamin Renoust, and Yuta Nakashima. **ContextNet: representation and exploration for painting classification and retrieval in context**. *International Journal of Multimedia Information Retrieval*, **9**(1):17–30, 2020.

[100] Giovanna Castellano, Giovanni Sansaro, and Gennaro Vessio. **Integrating contextual knowledge to visual features for fine art classification**. *arXiv preprint arXiv:2105.15028*, 2021.

[101] Giovanna Castellano, Vincenzo Digeno, Giovanni Sansaro, and Gennaro Vessio. **Leveraging Knowledge Graphs and Deep Learning for automatic art analysis**. *Knowledge-Based Systems*, **248**, 2022.

[102] Noa Garcia and George Vogiatzis. **How to read paintings: semantic art understanding with multi-modal retrieval**. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.